

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Leandro Simões da Silva

**MMRecommender: Arquitetura Aberta para
Sistemas de Recomendação**

JUIZ DE FORA, MG - BRASIL.
SETEMBRO 2017

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Leandro Simões da Silva

**MMRecommender: Arquitetura Aberta para
Sistemas de Recomendação**

Dissertação apresentada ao
Programa de Pós-Graduação em Ciência da
Computação, do Instituto de Ciências Exatas,
da Universidade Federal de Juiz de Fora como
requisito parcial para obtenção do título de
Mestre em Ciência da Computação.

Orientadora: Dra. Fernanda Cláudia Alves Campos.

JUIZ DE FORA, MG - BRASIL.
SETEMBRO 2017

Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Simões da Silva, Leandro.

MMRecommender : Arquitetura Aberta para Sistemas de Recomendação / Leandro Simões da Silva. -- 2017.
90 f.

Orientadora: Fernanda Claudia Alves Campos

Coorientador: Victor Stroele Menezes

Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, ICE/Engenharia. Programa de Pós-Graduação em Ciência da Computação, 2017.

1. Sistemas de Recomendação. 2. Arquitetura Aberta. 3. Engenharia de Software. I. Claudia Alves Campos, Fernanda, orient. II. Stroele Menezes, Victor, coorient. III. Título.

Leandro Simões da Silva

**MMRecommender: Arquitetura Aberta para
Sistemas de Recomendação**

Dissertação apresentada ao
Programa de Pós-Graduação em Ciência da
Computação, do Instituto de Ciências Exatas
da Universidade Federal de Juiz de Fora como
requisito parcial para obtenção do título de
Mestre em Ciência da Computação.

Aprovada em 1º de setembro de 2017.

BANCA EXAMINADORA

Profa. Fernanda Claudia Alves Campos, D. Sc. - Orientadora
Universidade Federal de Juiz de Fora

Prof. Victor Ströele Menezes, D. Sc. - Co-orientador
Universidade Federal de Juiz de Fora

Prof. Mario Antônio Ribeiro Dantas, PhD.
Universidade Federal de Santa Catarina

Profa. Regina Maria Maciel Braga Villela, D. Sc.
Universidade Federal de Juiz de Fora

JUIZ DE FORA, MG - BRASIL
SETEMBRO 2017

Dedico este trabalho ao meu pai, pelos ensinamentos
de vida que levo comigo. E por entender meus
sacrifícios para concluir o mestrado.

AGRADECIMENTOS

Agradeço aqueles que acreditaram em mim mesmo sem nenhum motivo... Estes sabem quem são.

Aos meus pais, biológicos ou não, que se sacrificaram para me darem oportunidades de estudo que eles não tiveram, ao apoio e cobrança que me fizeram seguir em frente.

Aos professores Saulo e Marco Antônio que me incentivaram a seguir para o mestrado...

À Professora Fernanda, pelos constantes ensinamentos, contribuições, cobranças, incentivos e principalmente, paciência, nessa caminhada. Ao co-orientador Victor, com suas contribuições e ensinamentos. E aos professores Regina, José Maria que me apoiaram quando pensei em desistir do mestrado. Aos professores Tiago Torrent e Ely Matos pela oportunidade de aplicar minha pesquisa em um cenário real.

Aos colegas de turma, que compartilharam conhecimentos e memes... Aos colegas de trabalho pela amizade, apoio e confiança no meu trabalho.

E a CAPES e a Universidade Federal de Juiz de Fora.

Obrigado.

*“Há doutos que não são doutores e há doutores que não
são doutos”*

- Professor Zeferino Vaz

RESUMO

Sistemas de Recomendação podem ser definidos como sistemas capazes de recomendar recursos aderentes ao perfil e contexto do usuário ou grupo de usuários, podendo ser aplicados em diversos domínios, tais como educação, turismo e *e-Science*. Devido a esta característica adaptável é possível encontrar diversos modelos de recomendação na literatura, cada um com combinações de métodos e algoritmos distintos. Essa variedade de modelos de recomendação pode dificultar o processo de implementação de Sistemas de Recomendação. Neste cenário, a presente dissertação apresenta a arquitetura aberta MMRecommender, onde através da combinação de componentes presentes em cada etapa é possível instanciar modelos de recomendação que podem ser aplicados a diversos domínios. Para avaliar a arquitetura são apresentados três estudos de casos sob perspectivas diferentes: o primeiro estudo de caso foca na adaptação de um Sistema de Recomendação existente para a arquitetura MMRecommender, o segundo estudo de caso implementa um modelo de recomendação criado a partir da arquitetura proposta em um ecossistema de software científico, e, por último, um estudo de caso evidenciando como a arquitetura proposta viabilizou a implementação de um Sistema de Recomendação turístico utilizado nas olimpíadas RIO 2016. Após a avaliação de cada estudo de caso foram obtidos indícios de que a arquitetura proposta pode auxiliar na construção de modelos de recomendação.

Palavras-chave: Sistemas de recomendação, modelos de recomendação, arquitetura aberta

ABSTRACT

Recommender Systems can be defined as systems capable of recommending resources adhering to user or group of user's profile and context, and can be applied in several domains, such as education, tourism and e-science. Due to this adaptive feature, it is possible to find several recommender models in the literature, each with combinations of different methods and algorithms. This variety of recommendation models can make it difficult to implement Recommender Systems. In this scenario, the present dissertation presents an open architecture MMRecommender, where through the combination of components present in each step it is possible to instantiate recommender models that can be applied to several application domains. To evaluate the architecture, three case studies are presented under different perspectives: the first case study focuses on the adaptation of an existing Recommender System to the MMRecommender architecture, the second case study implements a recommender model created from the proposed architecture in a scientific software ecosystem, and finally a case study evidences how the proposed architecture made possible the implementation of a Tourist Recommender System used in the RIO 2016 Olympic Games. After evaluating each case study it was possible to identify indications that the proposed architecture can help in the construction of recommender models.

Keywords: *Recommender systems, recommender models, open architecture.*

LISTA DE FIGURAS

Figura 1 - Diagrama de organização do trabalho.	20
Figura 2- Fluxo de trabalho evidenciando os trabalhos previamente selecionados após cada critério de inclusão e exclusão.....	31
Figura 3 - Arquitetura conceitual MMRecommender.	40
Figura 4 - Instância derivada do MMRecommender para o BROAD-GRS.....	46
Figura 5 - Recomendação de recursos educacionais do repositório BROAD para o Grupo 1.	46
Figura 6 - Arquitetura do E-SECO.	52
Figura 7- Módulo de Recomendação do E-SECO.....	53
Figura 8 - Serviços utilizados no E-SECO inseridos no plano cartesiano <i>rating e time</i>	57
Figura 9 - Ponto geométrico dos serviços utilizados no E-SECO.....	57
Figura 10 - Novo plano cartesiano com a média geométrica e os serviços retornados na busca.	57
Figura 11 - Fluxo de recomendação do E-SECO.	58
Figura 12 - Arquitetura de recomendação do E-SECO.	58
Figura 13 - Tela com recomendação de locais turísticos e detalhes do local.....	62
Figura 14 – Arquitetura do Mknob.....	62
Figura 15 - Arquitetura MMRecommender do aplicativo Mknob.	64
Figura 16 – Fluxo do aplicativo evidenciando as etapas e APIs usadas no processo de recomendação.	66
Figura 17 – Etapas do processo de recomendação usando na notação BPMN.	66
Figura 18 – Algoritmo em PHP dos métodos nGrams e Trigrams.....	70
Figura 19 – Algoritmo em PHP que calcula a similaridade de cosseno.....	70
Figura 20 – Algoritmo em PHP que calcula a distância Euclidiana.....	70
Figura 21 – Modelo entidade relacionamento do banco de dados utilizado no Mknob.....	71
Figura 22 – Lista de permissões aprovadas pelo <i>Facebook</i>	71

LISTA DE GRÁFICOS

Gráfico 1 - Quantidade de trabalhos aceitos por veículos de publicação com ocorrências maiores ou iguais a dois.	32
Gráfico 2 - Quantidade de trabalhos aceitos por ano de publicação.	33
Gráfico 3 - Quantidade de trabalhos por métodos.	33
Gráfico 4 - Técnicas extraídas dos trabalhos selecionados.	34
Gráfico 5 - Domínios de aplicação.	35
Gráfico 6 - Percentual de problemas recorrentes na área de sistemas de recomendação.	35
Gráfico 7 - Percentual das métricas de avaliação utilizadas nos trabalhos selecionados.	35
Gráfico 8 - Aceitação das características extraídas dos Grupos.	49
Gráfico 9 - Aceitação dos usuários em relação às recomendações feitas ao Grupo 1.	49
Gráfico 10 - Quantidade de usuários ativos por dia.	76
Gráfico 11 - Total de dispositivos de acesso por dia.	76
Gráfico 12 - Representação gráfica do total de usuários ativos e ações positivas por dia.	78
Gráfico 13 - Quantidade de pesquisas textuais realizadas no aplicativo por dia.	78
Gráfico 14 - Perfil de comportamento com base nas pesquisas textuais.	79
Gráfico 15 - Perfil de comportamento com base nos 'likes'.	79
Gráfico 16 - Buscas mais frequentes.	79
Gráfico 17 - Percentual de idiomas.	80
Gráfico 18 - Percentual por país.	80
Gráfico 19 - Histograma de idade.	80
Gráfico 20 - <i>Boxplot</i> por idade.	80

LISTA DE TABELAS

Tabela 1- GQM.....	28
Tabela 2- PICOC.	29
Tabela 3- <i>String</i> genérica derivada do PICOC.	29
Tabela 4 - Comparação entre os trabalhos relacionados.	36
Tabela 5 - Comparação do tempo de vida.	54
Tabela 6 - Escala de pontuação com base no desvio padrão.	55
Tabela 7 - Comparação das listas geradas com e sem a recomendação	60
Tabela 8 - Exemplo de vetor de característica representando quatro características: Compras, <i>Fast-food</i> , Café e <i>Rugby</i>	67
Tabela 9 - Aplicação do filtro inclusivo e exclusão do objeto 01.	67
Tabela 10 - Vetor de característica formado com os dados geográficos.*	67
Tabela 11 - Resultado do modelo de recomendação.	75
Tabela 12 - Quantidade de usuários ativos e ações positivas por dia.....	77
Tabela 13 - Relação país de origem e idioma principal.	81

SUMÁRIO

1. Introdução	15
1.1 Problemas.....	17
1.2 Questão de Pesquisa.....	17
1.3 Metodologia	18
1.4 Objetivo	18
1.5 Organização da Dissertação	19
2. Referenciais Teóricos	21
2.1 Métodos de Filtragem	21
2.1.1 Filtragem Baseada em Conteúdo.....	22
2.1.2 Filtragem Colaborativa	23
2.1.3 Filtragem Demográfica.....	24
2.1.4 Filtragem Social.....	24
2.1.5 Filtragem Híbrida	24
2.2 PERFIL E CONTEXTO	25
2.3 Considerações Finais do Capítulo.....	25
3. Mapeamento Sistemático	27
3.1 Metodologia de Pesquisa	27
3.2 Mapeamento Sistemático	28
3.3 Estratégias de Buscas.....	29
3.3.1 Fontes de Pesquisa.....	30
3.3.2 Critérios de Inclusão e Exclusão	30
3.3.3 Coleta de Dados.....	31
3.4 Relatório do Mapeamento Sistemático	31
3.5 Ameaças a Validade.....	34
3.6 Trabalhos Relacionados	36
3.7 Considerações Finais do Capítulo.....	37
4. MMRecommender: Proposta de uma Arquitetura Aberta para SR.....	39
4.1 Considerações Finais do Capítulo.....	41
5. Avaliação da Arquitetura MMRecommender	43
5.1 Avaliando a Arquitetura para um SR Educacional para Grupos	44
5.1.1 BROAD-GRS	45
5.1.2 Estudo de Caso I.....	46
5.1.3 Evidências Observadas	50
5.2 Avaliando a Arquitetura para um Ecossistema de Software.....	50
5.2.1 Ecossistema E-SECO	51
5.2.2 Estudo de Caso II.....	53
5.2.2.1 Identificação de Parâmetros para Recomendação de Serviços Científicos.....	54
5.2.2.2 Tempo de vida.....	54
5.2.2.3 Fatores de Similaridade	55
5.2.2.4 Modelo de Recomendação.....	56
5.2.3 Avaliação	59
5.2.4 Evidências Observadas	60
5.3 Avaliando a Arquitetura para um Sistema de Recomendação Turístico	61
5.3.1 Arquitetura do Sistema MKNOB-RECOMMENDER.....	63
5.3.1.1 Modelo de Recomendação.....	63
5.3.1.2 Processo de Recomendação	65

5.3.1.3	<i>Comparação Textual</i>	68
5.3.2	Desenvolvimento do Sistema	69
5.3.2.1	<i>Integração com Facebook</i>	71
5.3.2.2	<i>Coleta de Dados</i>	72
5.3.2.3	<i>Fonte de Dados</i>	73
5.3.3	Estudo de Caso III	73
	Evidências Observadas na Etapa 2	81
5.3.4	Considerações Finais da Seção.....	82
6	CONSIDERAÇÕES FINAIS	83
6.1	Contribuições	83
6.2	Limitações.....	84
6.3	Trabalhos Futuros	84
	REFERÊNCIAS BIBLIOGRÁFICAS	86
	APÊNDICE A	89
	Termos extraídos do mapeamento sistemático	89
	APÊNDICE B	90
	Trabalhos publicados e relação qualis	90

1. INTRODUÇÃO

A partir da década de 90 Sistemas de Recomendação (SR) começaram a ser estudados como uma área independente de pesquisa, com o foco na recomendação de itens não visualizados por usuários. Em termos gerais o problema de recomendar concentra esforços em estimar uma classificação para itens não visualizados considerando a classificação do próprio usuário para outros itens, ou informações fornecidas ao sistema (ADOMAVICIUS; TUZHILIN, 2005).

Sistemas de recomendação podem ser definidos como sistemas capazes de recomendar objetos virtuais de acordo com o perfil do usuário, ou guiá-lo de forma personalizada para esses objetos virtuais (BURKE, 2002) e se aplicam a domínios como *e-commerce*, *e-learning*, conteúdo multimídia, turismo, entre outros. Essa característica adaptável a diversos domínios faz com que a área de SR seja um campo fértil para diversos trabalhos acadêmicos ou aplicações na indústria, conforme observado por (ADOMAVICIUS; TUZHILIN, 2005) e posteriormente por (BOBADILLA et al., 2013).

Como evidenciado por (NÚÑEZ-VALDÉZ et al., 2012) o processo de guiar o usuário até objetos virtuais aderentes ao seu perfil pode evitar o problema de sobrecarga de informação presente em alguns ambientes virtuais, tal problema é observado pelo autor como uma característica a ser considerada devido ao aumento de informações disponíveis na *Internet* e também como um problema de otimização a ser tratado na área de SR.

Um Sistema de Recomendação sugere recursos ao usuário baseado em suas preferências e expectativas (CAZELLA et al., 2009). Esses sistemas são capazes de recomendar produtos, serviços e objetos alinhados ao perfil e contexto do usuário ou grupo. Em pesquisas anteriores do grupo destacam-se os trabalhos de (ALMEIDA; CAMPOS; STROELE, 2015) e (SIMOES et al., 2016) sugerindo diversos fatores que combinados definem um sistema de recomendação personalizado: tipos de recursos a serem recomendados, algoritmos de filtragem, métodos, técnicas, modelos de extração e enriquecimento de perfil e contexto, desempenho e qualidade dos resultados para o processo de recomendação personalizado. Entretanto, também é necessário utilizar de algoritmos para efetuar a otimização e classificação dos dados, conforme demonstrado por (TIAN et al., 2014a), existe uma série de algoritmos que podem oferecer resultados diferentes de acordo com a natureza dos dados utilizados para a recomendação.

Em ambientes sociais os SR são de particular importância, porque os usuários compartilham recursos formando grupos com interesses comuns (REZENDE et al., 2015) e (RIBEIRO; FONSECA; FREITAS, 2013). Nesse cenário de interação um dos grandes desafios desses sistemas é como lidar adequadamente com as preferências de cada integrante de um grupo para geração de uma recomendação conjunta (CARVALHO; MACEDO, 2014).

No processo de recomendação são necessários três componentes, sendo eles: os dados de contexto e perfil do usuário e algoritmos de recomendação (BURKE, 2002). A etapa de entender o contexto e definir o perfil do usuário é muito importante (JAWAHEER; SZOMSZOR; KOSTKOVA, 2010a), pois o processo de extração de informações está muito influenciado pelo crescimento da *Internet* e redes sociais. Neste contexto é possível encontrar trabalhos que possuem o foco de recomendação com base em redes sociais, podendo recomendar objetos virtuais ou utilizar os dados de perfil e contexto provenientes desta fonte (PEREIRA et al., 2014) e (ALMEIDA et al., 2015).

Este trabalho pretende avançar as pesquisas relacionadas ao projeto BROAD de recomendação de recursos educacionais (ALMEIDA; CAMPOS; STROELE, 2015), (NERY et al., 2012), (PEREIRA et al., 2014), (PEREIRA et al., 2015) e (REZENDE et al., 2015). A principal idéia do projeto BROAD é que cada nova versão da arquitetura represente um passo à frente em relação aos objetivos da pesquisa considerando a adoção de novas tecnologias. Entretanto, o foco deste trabalho é o reuso de soluções para recomendações individuais ou em grupo, usando repositórios locais, dados ligados ou outras mídias disponíveis, como, por exemplo, redes sociais.

Existem diferentes estratégias para o desenvolvimento intensivo de *software*, uma dessas abordagens é o desenvolvimento usando arquitetura aberta (ALSPAUGH; ASUNCION; SCACCHI, 2013). Essa abordagem de reuso permite a redução de custos, amplia a confiabilidade e funcionalidade do sistema e acelera o processo de desenvolvimento de *software* para atender requisitos específicos.

“Arquitetura aberta é uma técnica de customização introduzida por (OREIZY, 2000 aput ALSPAUGH et al., 2013), que permite que terceiros modifiquem o sistema de *software* através de uma arquitetura pré-definida, evoluindo o sistema através da recolocação de seus componentes”. A proposta dessa dissertação é uma arquitetura de alto nível para sistemas de recomendação que inclui os principais módulos e elementos, e que pode ser instanciada para diferentes configurações e domínios.

1.1 PROBLEMAS

No contexto de sistemas de recomendação existem diversas abordagens com métodos, técnicas e algoritmos que combinados criam modelos de recomendações diferentes, e, com isto, distintos modelos arquiteturais podem representar esses SR. Deste cenário surge o primeiro problema tratado neste trabalho, que é definido como: É possível propor uma arquitetura que pode ser instanciada para a representação de SR já existentes?

Sistemas de recomendação podem ser adaptados para diversos domínios de aplicação, e podem integrar aplicações existentes, através de um módulo acoplado ao sistema ou uma camada de serviço que forneça dados para o processamento da recomendação e consuma o resultado apresentando para o usuário. Nesse cenário surge o segundo problema explorado neste trabalho que pode ser assim definido: É possível instanciar uma arquitetura de recomendação adaptável que auxilie o processo de implementação de módulos ou camadas de serviços de recomendação em sistemas já existentes?

Descobrir objetos virtuais aderentes ao perfil e contexto do usuário também pode ser o objetivo de sistemas de informação, neste contexto tais sistemas têm como principal objetivo auxiliar a descoberta de novos conteúdos relevantes para usuários ou grupos de usuários, sendo este o terceiro problema tratado: A partir de uma arquitetura aberta é possível instanciar sistemas de recomendação com métodos, algoritmos, dados de perfil e contexto bem definidos?

1.2 QUESTÃO DE PESQUISA

Sistemas de recomendação podem ser aplicados a diferentes domínios utilizando diferentes métodos e algoritmos a partir de diferentes tipos de perfil e contexto de usuários, desta forma foi definida a questão de pesquisa que norteou e motivou a execução do presente trabalho. Esta questão de pesquisa pode ser enunciada como:

- É possível propor uma arquitetura aberta que pode ser instanciada gerando diferentes modelos de recomendação, com abordagens, métodos e algoritmos, aplicados em diversos domínios com base em perfis e contextos de usuários ou grupos de usuários diversificados, a fim de garantir a assertividade do sistema de recomendação?

1.3 METODOLOGIA

A metodologia utilizada neste trabalho pode ser dividida em três etapas, sendo elas: O processo de **Mapeamento Sistemático da Literatura** (SIMOES et al., 2016), em seguida os resultados foram comparados com conhecimentos prévios do grupo de pesquisa, na etapa de **Validação**, e por fim, a etapa de **Aplicação** onde a arquitetura proposta foi implementada em diferentes domínios de aplicação, sob diferentes perspectivas.

O mapeamento sistemático foi importante para identificar o estado da arte e prática no contexto de sistemas de recomendação que podem ser aplicados a diferentes domínios e identificar métodos e algoritmos utilizados na área. Também foi possível identificar termos comuns e autores relevantes ao meio, além de levantar trabalhos que oferecem definições importantes para a área estudada.

Na etapa de validação itens relevantes levantados na etapa anterior foram comparados com as experiências do grupo de pesquisa a fim de identificar possíveis itens importantes que não foram evidenciados nos resultados do mapeamento. Ao final desta etapa foram gerados os primeiros esboços da arquitetura proposta, categorizando as etapas, com métodos e algoritmos, necessários para o processo de recomendação incluindo os modelos identificados na literatura. Esse modelo passou por um processo de validação incremental, onde itens importantes foram adicionados a partir da experiência, conhecimentos prévios e validações externas, gerando a arquitetura MMRecommender.

Por fim, a arquitetura foi aplicada em sistemas de recomendação em domínios distintos, sendo *e-learning*, *e-Science* e turístico. Cada aplicação foi feita como um estudo de caso com características e perspectivas distintas quanto à avaliação da arquitetura.

1.4 OBJETIVO

Diante da diversidade de algoritmos de filtragem, métodos e técnicas, modelos de extração e enriquecimento de perfil e contexto, e, principalmente, da diversidade de domínios de aplicação e recursos recomendáveis nos SR, o objetivo principal desta dissertação é fornecer uma arquitetura aberta e adaptável para a construção ou instanciação de sistemas de recomendação, levando em consideração os diversos domínios em que são utilizados.

1.5 ORGANIZAÇÃO DA DISSERTAÇÃO

Neste capítulo foram apresentadas as motivações que justificam a proposta dessa dissertação, as questões de pesquisa e metodologias que guiaram todo o processo de pesquisa e desenvolvimento aqui descrito, além da definição do objetivo e a organização do presente trabalho. No Capítulo 2 serão abordados os pressupostos teóricos. O Capítulo 3 descreve as etapas do mapeamento sistemático da literatura e apresenta uma análise dos resultados obtidos além dos trabalhos que serviram como base na construção do conhecimento prévio para iniciar os estudos e oferecer uma contribuição para a área de sistemas de recomendação. Em sequência, no Capítulo 4, é apresentada a arquitetura proposta neste trabalho, denominada de MMRecommender, além de descrever os componentes intrínsecos do modelo. A avaliação da arquitetura está descrita no Capítulo 5, apresentando três estudos de caso distintos de aplicação da arquitetura em três domínios: *e-learning*, *e-Science* e turismo. Cada estudo de caso possui suas particularidades e objetivos específicos. Por fim, o Capítulo 6, apresenta as considerações finais e sugestões de trabalhos futuros. A Figura 1 é uma síntese da organização desta dissertação.

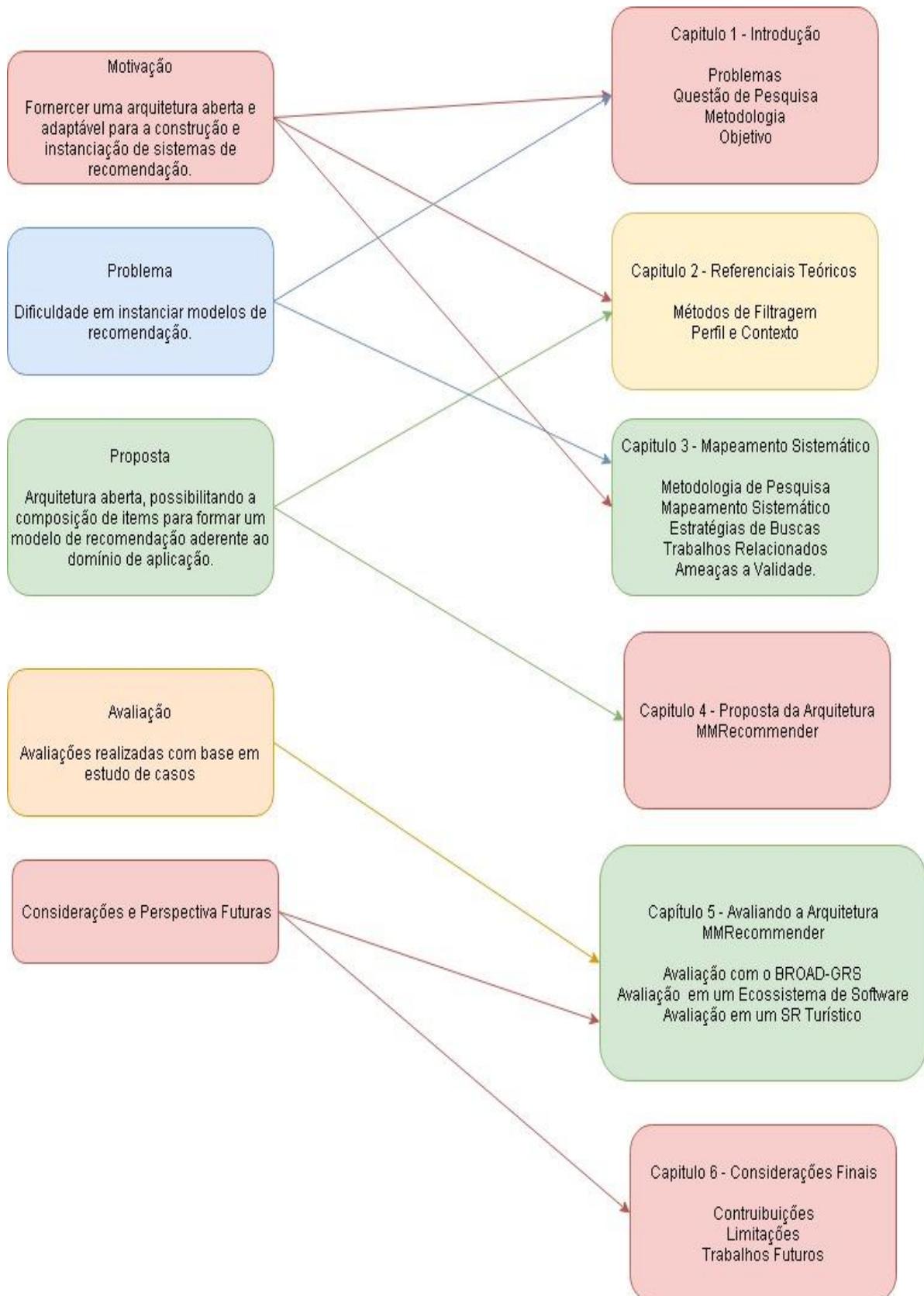


Figura 1 - Diagrama de organização do trabalho.

2. REFERENCIAIS TEÓRICOS

Neste capítulo, são apresentados os principais referenciais teóricos que contribuíram para a construção do presente trabalho.

Sistemas de recomendação aplicados a diversos domínios têm a finalidade de resolver problemas distintos. Devido a essa característica é possível encontrar na literatura trabalhos em SR com diversas finalidades, podendo ser trabalhos que aplicam a recomendação como fim, ou seja, trabalhos com foco na evolução de modelos ou algoritmos de recomendação, e trabalhos que aplicam a recomendação como meio, isto é, trabalhos que utilizam a recomendação como apoio à função principal do sistema.

Neste contexto é possível citar dois sistemas do mesmo domínio de aplicação que utilizam a recomendação como fim e como meio, o primeiro deles é o site *MovieLens* (www.movielens.org) que utiliza a recomendação como fim para recomendar filmes, diferenciando da plataforma *Netflix* (www.netflix.com), que tem como objetivo a exibição de filmes, neste caso a *Netflix* utiliza a recomendação como meio para auxiliar o usuário na identificação de filmes aderentes ao seu perfil.

Em seguida conceitos importantes sobre arquitetura aberta, características e vantagens são abordados com o foco na aplicação em sistemas de recomendação, mais especificamente na construção da arquitetura proposta nas seções posteriores deste trabalho.

2.1 MÉTODOS DE FILTRAGEM

Segundo (ADOMAVICIUS; TUZHILIN, 2005) é possível classificar SR em três grupos definidos a partir do método de filtragem empregado, sendo **filtragem baseada em conteúdo**, **filtragem colaborativa** e **filtragem híbrida**. O método de filtragem pode ser resumido como um processo que separa os objetos aderentes ao perfil e contexto do usuário dos demais objetos que não estão aderentes ao perfil e contexto do usuário. Desta forma existem métodos que são mais assertivos que outros dependendo do domínio da aplicação e em como os objetos são descritos em termos computacionais.

O trabalho apresentado por (BOBADILLA et al., 2013) considera válida a classificação de SR a partir do método de filtragem empregado e também introduz outros dois métodos encontrados na literatura, o primeiro deles é a **filtragem demográfica**, e, por fim, é apresentada a **filtragem social** como uma possível tendência na área.

2.1.1 Filtragem Baseada em Conteúdo

Este método de filtragem compara o quão similar são os objetos que podem ser recomendados com as características do usuário que irá receber a recomendação, como, por exemplo, um sistema de recomendação para filmes, que utiliza filtragem baseada em conteúdo, busca por filmes com características similares como gênero, atores, diretor, entre outros, aderentes ao perfil do usuário. Conforme descrito por (ADOMAVICIUS; TUZHILIN, 2005) este método tem origem no campo de recuperação e filtragem de informação, amplamente utilizados em sistemas baseados na filtragem de texto e foi introduzido em SR devido à facilidade em lidar com diferentes objetos que possuem descrição textual, como *websites*, documentos, mensagens, entre outros.

Métodos baseados em filtragem por conteúdo utilizam métricas de similaridade para definir os objetos que serão recomendados ao usuário. Na literatura é possível encontrar diversos algoritmos com este objetivo, entre eles: Similaridade de Cosseno (CHEN, 2012), Distância Euclidiana (YANG; LI, 2007) e Jaccard (GAO; ZHANG, 2014). Estes algoritmos são de fácil implementação e carregam conceitos matemáticos simples e produzem resultados satisfatórios, por isso a filtragem baseada em conteúdo é amplamente utilizada, porém este método também possui desvantagens como o *overspecialization* e partida fria (*cold-start*).

O primeiro problema, *overspecialization*, é identificado quando um SR não consegue sugerir novos objetos ao usuário, isso ocorre porque o método utiliza algoritmos de similaridade para buscar objetos que possuem maior similaridade com o perfil e contexto do usuário. Desta forma o método ignora objetos que, de certa forma, também são aderentes ao usuário, porém não possuem altos índices de similaridade. Este problema pode ser exemplificado em um sistema de músicas onde o usuário pode escolher diferentes gêneros musicais, neste caso o usuário prefere Blues e Rock, porém, por qualquer motivo, este usuário avalia mais músicas de Rock do que Blues, devido à característica do método, este sistema irá recomendar apenas músicas do gênero Rock, mesmo que o usuário também tenha avaliado algumas músicas de Blues, mas durante o processo de cálculo de similaridade as músicas de Rock estarão com maiores índices de similaridade.

O problema da partida fria é amplamente estudado em SR e de forma resumida está relacionado ao que recomendar quando não existem informações do usuário. Como o método baseado em conteúdo utiliza das escolhas passadas do usuário ele é altamente dependente destas informações históricas, logo sem essas informações o método não consegue fazer

recomendações iniciais.

Na literatura é possível encontrar trabalhos que propõem métodos para mitigar os problemas descritos anteriormente, como a filtragem colaborativa que oferece outra abordagem.

2.1.2 Filtragem Colaborativa

Como descrito em (BOBADILLA et al., 2013) o método de filtragem colaborativa filtra objetos com base nas avaliações de outros usuários com preferências semelhantes ao usuário que irá receber a recomendação. Um exemplo de filtragem colaborativa é a recomendação de grupos e páginas do *Facebook*, que tende a recomendar páginas e grupos para um usuário com base na quantidade de amigos presentes naquela página ou grupo. Deste modo a filtragem colaborativa consegue inferir uma possível preferência do usuário com base nas preferências já confirmadas de outros usuários similares.

Alguns dos algoritmos comumente utilizados para realizar a filtragem colaborativa são *knn* (*k nearestneighbors*) (QIAN et al., 2013) e *matrix factorization* (ZHANG; LIU, 2015). Estes algoritmos são de fácil entendimento e implementação abstraída através de *APIs* de *machine learning* como *Apache Mahout* (mahout.apache.org).

Esta abordagem resolve o problema de *overspecialization*, presente na abordagem baseada em conteúdo, porém permanece o problema da partida fria no contexto dos objetos que podem ser recomendados, ou seja, para novos objetos inseridos no sistema que ainda não foram avaliados. Outro problema enfrentado pela abordagem colaborativa é amplamente discutido em trabalhos da área e conhecido como *sparsity problem*.

Sparsity problem pode ser descrito como a dificuldade em lidar com dados esparsos da base de dados utilizada pelo SR. Esse problema ocorre devido a uma característica de SR que possui poucos objetos avaliados, ou menos do que o necessário. Neste caso o algoritmo utilizado precisa percorrer muitos objetos que não foram avaliados ou possuem poucas avaliações, por isso a assertividade de SR baseados na colaboração é altamente dependente de um grande volume de usuários.

2.1.3 Filtragem Demográfica

Segundo (BOBADILLA et al., 2013) a filtragem demográfica se baseia no conceito de que indivíduos de um determinado grupo possuem preferências similares, este grupo pode ser descrito por gênero, religião, idade, entre outros fatores. A princípio essa abordagem pode ser confundida com a abordagem colaborativa, porém a diferença está que na abordagem demográfica as preferências de um determinado grupo são conhecidas previamente. Tais preferências podem ter origem em pesquisas demográficas ou viés, como por exemplo um SR que recomenda filmes pode recomendar filmes de ação para usuários do gênero masculino por considerar que homens tendem a escolher filmes de ação.

2.1.4 Filtragem Social

A filtragem social utiliza como base as informações sociais do usuário, que podem ser capturadas de forma explícita ou implícita, formando redes de afinidade entre usuários, tais redes também podem ser formadas de forma explícita ou implícita. Em (BOBADILLA et al., 2013) informações sociais são definidas como dados gerados com a finalidade de comunicação entre usuários, como mensagens e *posts* em *blogs* ou redes sociais.

2.1.5 Filtragem Híbrida

Na literatura são encontrados diversos trabalhos que implementam uma abordagem híbrida em SR, como visto em (BADARO et al., 2013; LEI et al., 2012; LUCAS et al., 2013; ULLAH et al., 2012; WEN; FANG; GUAN, 2012). Os autores (BOBADILLA et al., 2013) definem a filtragem híbrida como um método que implementa parcialmente ou totalmente dois ou mais tipos de filtragens. Em (ADOMAVICIUS; TUZHILIN, 2005) os autores apresentam uma proposta de junção das filtragens colaborativas e baseadas em conteúdo de modo cooperativo, sendo uma das primeiras abordagens de filtragem híbrida. Esta abordagem foi utilizada para mitigar alguns problemas não resolvidos pela filtragem baseada em conteúdo e a filtragem colaborativa, além de unir os pontos positivos de cada abordagem.

2.2 PERFIL E CONTEXTO

Conforme evidenciado por (JAWAHEER; SZOMSZOR; KOSTKOVA, 2010a) para realizar a recomendação é necessário identificar o perfil e contexto do usuário ou grupo que irá receber a recomendação, neste processo são usados dados extraídos ou inseridos no sistema pelo usuário. Desta forma, é possível categorizar o perfil do usuário em dois aspectos, sendo o perfil implícito e explícito.

O perfil implícito é criado a partir dos dados implícitos, que podem refletir o comportamento do usuário no sistema, como páginas curtidas, filmes assistidos, entre outros, como mencionado por (JAWAHEER; SZOMSZOR; KOSTKOVA, 2010b). Os dados implícitos são mais abundantes dentro de um sistema, porém podem não refletir as preferências reais do usuário. Por outro lado, o perfil explícito reflete um grau de certeza maior quanto às preferências do usuário, porém são mais escassos, pois dependem da colaboração do usuário ao avaliar os itens mais aderentes ao seu perfil. Na literatura podemos encontrar trabalhos que medem as alterações de preferências do usuário, usando dados explícitos e implícitos, essa abordagem será mais explorada em um dos estudos de caso apresentados neste trabalho.

Sob o ponto de vista do contexto existem diversas definições presentes na literatura, porém o objetivo deste trabalho não é apresentar uma nova definição para contexto, por isso foi utilizada a definição apresentada por (FLEISHMANN; BASTOS; PERNAS, 2012) que está aderente ao escopo deste trabalho. De forma resumida é possível definir o contexto como qualquer informação inserida ou capturada pelo sistema que possa influenciar na sua usabilidade. Desta forma o contexto é categorizado em duas dimensões, sendo a dimensão interna, relativa ao usuário do sistema, atributos como preferência, objetivos, competências, entre outros, e a dimensão externa, referente às características do ambiente, localização ou temporal.

2.3 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou os referenciais teóricos abordando os principais termos e definições da área, como tipos de filtragem, que são usados para categorizar SR, definições de perfis e contexto e apresentou alguns dos problemas comuns na área de SR. No capítulo posterior são

apresentados detalhes do mapeamento sistemático, com protocolo de busca, critérios de inclusão e exclusão, por fim é apresentado uma análise dos artigos selecionados para o mapeamento sistemático.

3. MAPEAMENTO SISTEMÁTICO

Com o objetivo de entender o estado da arte da área e identificar possíveis problemas ainda não explorados, foi conduzido um mapeamento sistemático com foco na busca de métodos, técnicas, arquiteturas e algoritmos utilizados em modelos ou sistemas de recomendação (SIMOES et al., 2016).

A pesquisa foi direcionada pela metodologia de mapeamento e revisão sistemática da literatura apresentado por (KITCHENHAM; CHARTERS, 2007) e descrito por (WOHLIN et al., 2012), onde são apresentadas as principais diferenças entre o mapeamento e a revisão sistemática. Como um dos objetivos deste trabalho é obter uma visão geral do estado da arte e estado da prática da área, como a sua evolução, problemas comuns, métodos e técnicas utilizados na academia e indústria, a metodologia de mapeamento sistemático foi mais aderente ao escopo da pesquisa.

3.1 METODOLOGIA DE PESQUISA

Seguindo os conceitos de (KITCHENHAM; CHARTERS, 2007) foi definido *a priori* o protocolo de pesquisa, dessa forma o mapeamento pode ser validado e replicado por profissionais e pesquisadores, além de fornecer um ponto de partida para uma extensão na pesquisa. Esta seção apresenta o protocolo do mapeamento sistemático. O estudo foi realizado entre agosto e dezembro de 2015.

O processo foi dividido em três etapas principais, sendo elas:

- Planejamento
 - Definição das questões de pesquisa;
 - Definição do protocolo de pesquisa;
- Condução
 - Seleção das bases indexadas;
 - Identificação dos trabalhos aderentes à pesquisa;
 - Extração de dados;
 - Consolidação e análise dos dados extraídos;
- Divulgação
 - Empacotamento dos dados gerados na pesquisa;
 - Divulgação dos resultados para a validação dos pares.

3.2 MAPEAMENTO SISTEMÁTICO

Norteados pelos conceitos de (WOHLIN et al., 2012) o método GQM (Tabela 1) ilustra o escopo da pesquisa em três aspectos, sendo eles:

- Conceitual, no qual são definidos os objetivos da pesquisa;
- Operacional, no qual são levantadas as perguntas que devem ser respondidas pelo mapeamento sistemático; e
- Quantitativo, no qual são definidas as métricas para avaliar os trabalhos identificados.

Método GQM	
<i>Goal</i>	Identificar modelos de recomendação.
<i>Question</i>	Quais as etapas necessárias na construção de sistemas de recomendação?
<i>Metric</i>	Quantidade de componentes em modelos de recomendação.

Tabela 1- GQM.

Após a construção do GQM foi definida a questão primária (QP) de pesquisa e a questão secundária (QS) de forma ampla, sem especificar um escopo fechado de pesquisa, como trabalhos que realizaram um experimento ou atuaram em um contexto específico. Definir questões de pesquisa menos restritivas é uma das características do mapeamento sistemático (WOHLIN et al., 2012).

- QP: Quais são as etapas necessárias na construção de sistemas de recomendação?
- QS: Quais as técnicas utilizadas na construção de modelos de recomendação?

Com essas questões espera-se identificar como um sistema pode gerar recomendações alinhadas ao perfil do usuário ou do grupo de usuários.

3.3 ESTRATÉGIAS DE BUSCAS

Para a busca dos possíveis trabalhos aderentes ao mapeamento foi utilizada a pesquisa em bases digitais indexadas utilizando *strings* de busca formadas por palavras-chaves e conectores lógicos. Na construção da *string* genérica foi utilizado o *framework* PICOC (*Population, Intervention, Comparison, Outcomes, Context*) proposto em (KITCHENHAM; CHARTERS, 2007). Uma breve pesquisa *ad-hoc* ajudou na descoberta dos termos mais usados e seus sinônimos, e, conseqüentemente, na construção do quadro PICOC.

PICOC		
Termo	Descrição	Keywords
<i>Population</i>	Algoritmos, abordagens e métodos baseados em modelo ou memória.	<i>memory-based, model-based, algorithms, approaches, methods</i>
<i>Intervention</i>	Efetuar recomendação	<i>Recommendation</i>
<i>Comparison</i>	--	--
<i>Outcome</i>	Modelos de recomendação	<i>Recommender model, recommendation model</i>
<i>Context</i>	Sistemas de recomendação	<i>recommender system, recommendation system</i>

Tabela 2- PICOC.

A partir do PICOC a *string* de busca genérica (Tabela 3) foi gerada e posteriormente adaptada às regras de busca de cada base digital definida no protocolo do mapeamento. O processo de construção do GQM, PICOC e *string* de busca foram validados por especialistas da área de experimentação em Engenharia de *Software*, sistemas de recomendação e, posteriormente, pela comunidade acadêmica em (SIMOES et al., 2016)

<i>String</i> genérica
<i>(“memory-based” OR “model-based” OR “algorithms” OR “approaches” OR “methods”) AND (“recommender model” OR “recommendation model”) OR (“recommender system” OR “recommendation system”)</i>

Tabela 3- *String* genérica derivada do PICOC.

3.3.1 Fontes de Pesquisa

Após a definição da string genérica foram escolhidas as bases de pesquisa, e uma *string* específica, derivada da *string* genérica, foi montada para cada base a fim de obter os resultados mais coerentes com os termos definidos no PICOC (Tabela 2).

- *Scopus* (www.scopus.com);
- *IEEE* (www.ieee.org);
- *Science@Direct* (www.sciencedirect.com);
- *ACM DL*(www.dl.acm.org);
- *Web of Science* (www.webknowledge.com).

As *strings* específicas foram criadas com base na documentação para buscas avançadas de cada base, em seguida os resultados obtidos foram analisados a fim de identificar possíveis falhas e divergências nos motores de busca. Esse processo é importante para garantir que diferentes *strings* realizem a mesma consulta lógica em todas as bases, mitigando possíveis problemas nas etapas posteriores.

3.3.2 Critérios de Inclusão e Exclusão

Os trabalhos retornados na etapa anterior foram submetidos aos critérios de inclusão:

- CI1: Artigos primários
- CI2: Trabalhos nos idiomas espanhol, inglês e português.

Além dos critérios de exclusão:

- CE1: Ano de publicação dos trabalhos, sendo excluídos os anteriores a 2005;
- CE2: Trabalhos duplicados foram excluídos mantendo sempre o mais recente;
- CE3: Trabalhos sem acesso ao texto completo;
- CE4: Trabalhos que não abordam o tema principal (modelos ou componentes de recomendação).

A Figura 2 mostra o fluxo de trabalho de cada etapa do mapeamento sistemático evidenciando os trabalhos previamente selecionados em cada critério.

3.3.3 Coleta de Dados

Com a documentação das etapas do mapeamento sistemático é possível gerar informações relevantes que, inicialmente, não foram definidas na etapa de planejamento. Com o auxílio de ferramentas como *Mendeley* (www.mendeley.com), *Parsifal* (parsif.al) e *MS Excel* (www.office.com), os trabalhos foram agrupados a fim de extrair informações relevantes, como os principais veículos de publicação (Gráfico 1), extraído a partir da quantidade total de trabalhos aceitos por veículo de publicação com ocorrência maior ou igual a dois.

Analisando a quantidade de trabalhos aceitos por ano de publicação (Gráfico 2) é possível identificar um crescimento nas publicações, sendo possível inferir que SR é um tema atual e de relevância para a comunidade acadêmica.

3.4 RELATÓRIO DO MAPEAMENTO SISTEMÁTICO

O mapeamento sistemático é conduzido de forma que as questões de pesquisa definidas nas etapas anteriores sejam respondidas. Nesta seção serão apresentadas as respostas às questões de pesquisa.

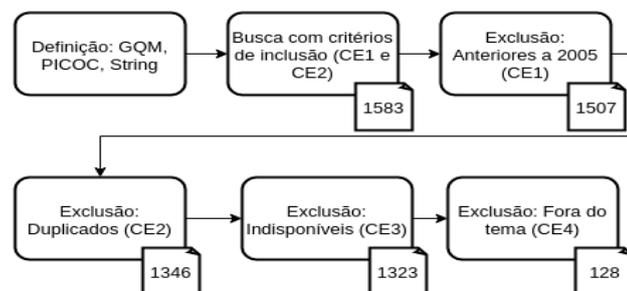


Figura 2- Fluxo de trabalho evidenciando os trabalhos previamente selecionados após cada critério de inclusão e exclusão.

Veículos de Publicação

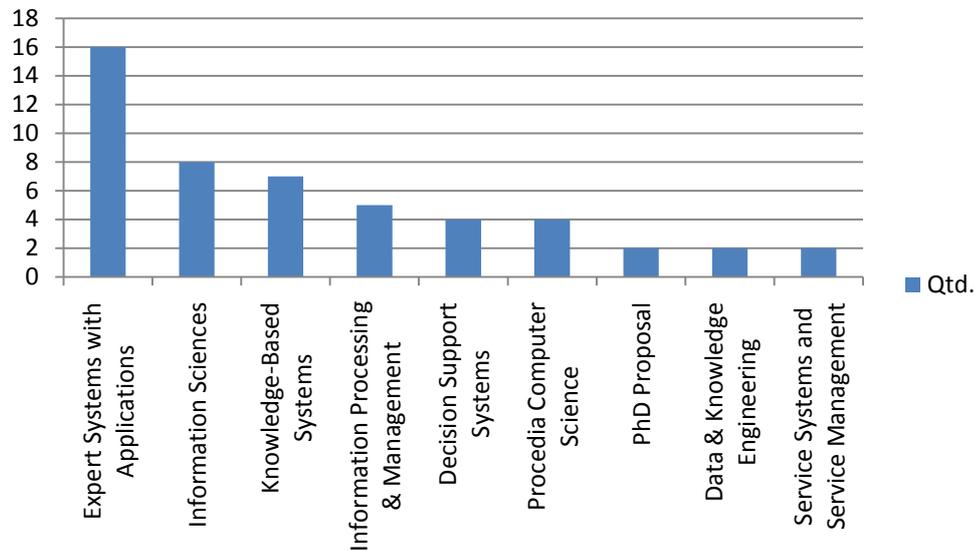


Gráfico 1- Quantidade de trabalhos aceitos por veículos de publicação com ocorrências maiores ou iguais a dois.

Para responder a QP - Quais são as etapas necessárias na construção de sistemas de recomendação? – foi realizada a extração de informações dos trabalhos selecionados, porém não foi possível identificar a existência de um consenso, *framework* ou modelo que pudesse responder a questão de pesquisa. Esse cenário forneceu evidências de que a arquitetura aberta, proposta neste trabalho, é relevante no sentido de oferecer um guia com etapas bem definidas para a construção de modelos de recomendação.

Para responder a QS - Quais as técnicas utilizadas na construção de modelos de recomendação? – os trabalhos foram agrupados com base nas definições de (ADOMAVICIUS; TUZHILIN, 2005; BOBADILLA et al., 2013) que dividem SR em cinco grupos, sendo eles: Filtragem Colaborativa, Filtragem Baseada em Conteúdo, Demográfica, Social e Híbrida. O Gráfico 3 apresenta o percentual agrupado dos trabalhos selecionados. É importante ressaltar que não foram identificados trabalhos que utilizam a filtragem social.

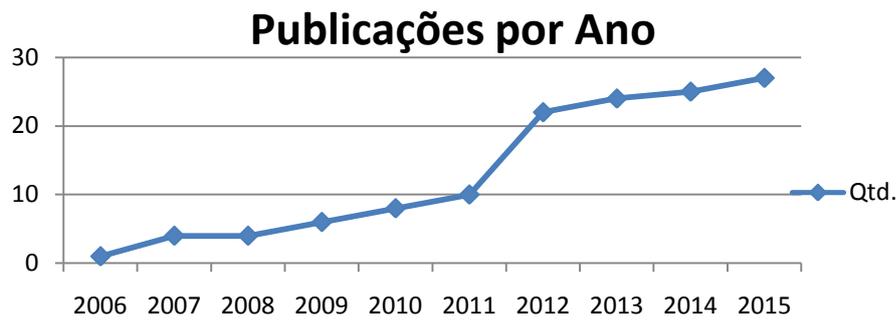


Gráfico 2 - Quantidade de trabalhos aceitos por ano de publicação.

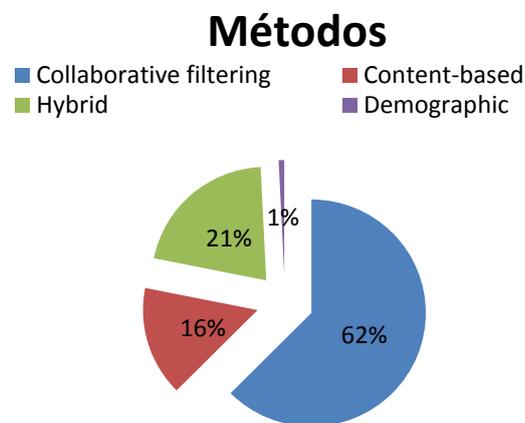


Gráfico 3 - Quantidade de trabalhos por métodos.

Em seguida os trabalhos foram agrupados pela técnica de recomendação aplicada, isso é, o algoritmo ou modelo que define quais objetos serão recomendados ao usuário ou grupo de usuários. Alguns trabalhos utilizaram mais de uma técnica para realizar a recomendação, nesse cenário o modelo de recomendação implementado foi analisado para extrair a técnica mais importante, ou seja, aquela que efetivamente define os objetos recomendados com base no perfil do usuário ou grupos de usuários. O Gráfico 4 ilustra as técnicas extraídas dos trabalhos selecionados.

Com a finalidade de entender a área de pesquisa foi realizada extração de dados não previstos no protocolo do mapeamento, dessa forma novos agrupamentos foram realizados possibilitando uma compreensão mais ampla sobre SR. O domínio de aplicação (Gráfico 5), isso é, onde o SR foi implementado, fornece evidências sobre onde SR são mais utilizados.

Conforme evidenciado por (BOBADILLA et al., 2013) existem problemas comuns na área de SR, como por exemplo, o problema da partida fria (*cold-start*) que foi o problema mais comum tratado entre os trabalhos. Muitos trabalhos não especificaram o tratamento de nenhum problema recorrente da área e por isso foram enquadrados no grupo “Não

especificado”. O Gráfico 6 apresenta o percentual dos problemas tratados pelos trabalhos.

A avaliação do modelo de recomendação é um fator importante para medir a eficiência do SR, por isso as métricas utilizadas foram agrupadas (Gráfico 4) a fim de identificar as métricas mais recorrentes. Alguns trabalhos (9%) não utilizaram nenhuma métrica de avaliação, sendo agrupados em “Não especificado”.

3.5 AMEAÇAS A VALIDADE

A validação do protocolo do mapeamento sistemático foi feita por um pesquisador da área, porém, não houve validação ou participação de um grupo de pessoas na seleção e extração dos dados dos trabalhos, sendo essa uma ameaça à validade do mapeamento sistemático devido à possível presença de viés.

Outra ameaça está ligada à quantidade de trabalhos analisados (1346) em um curto período de tempo, o que demandou grande esforço podendo ter ocasionado fadiga no processo.

Em relação ao processo de seleção dos trabalhos nas fontes de pesquisa, as *strings* de busca não foram exatamente as mesmas, logo, há a possibilidade de uma *string* específica não realizar a busca conforme esperado. Para mitigar esta ameaça as *strings* foram criadas com base nos tutoriais e arquivos de ajuda de cada base digital.

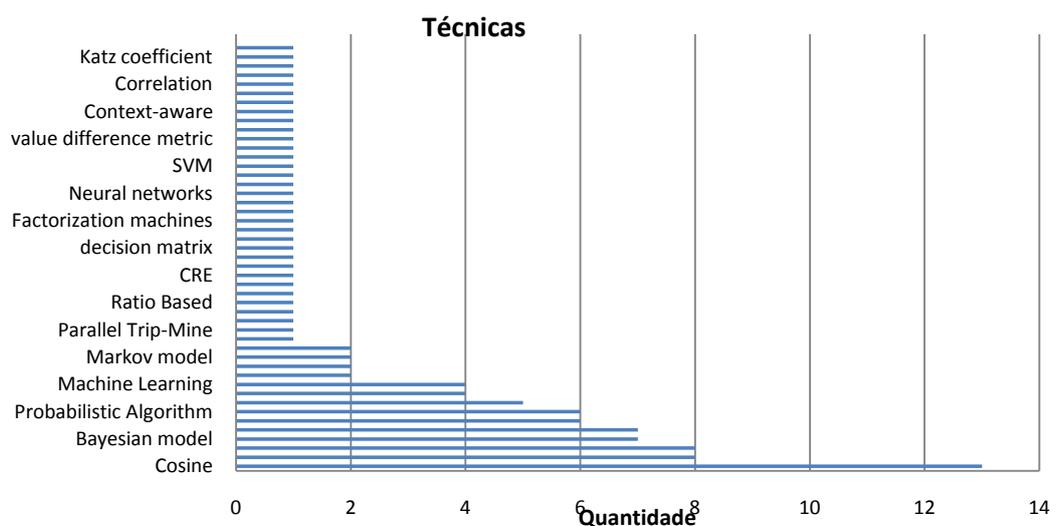


Gráfico 4 - Técnicas extraídas dos trabalhos selecionados.

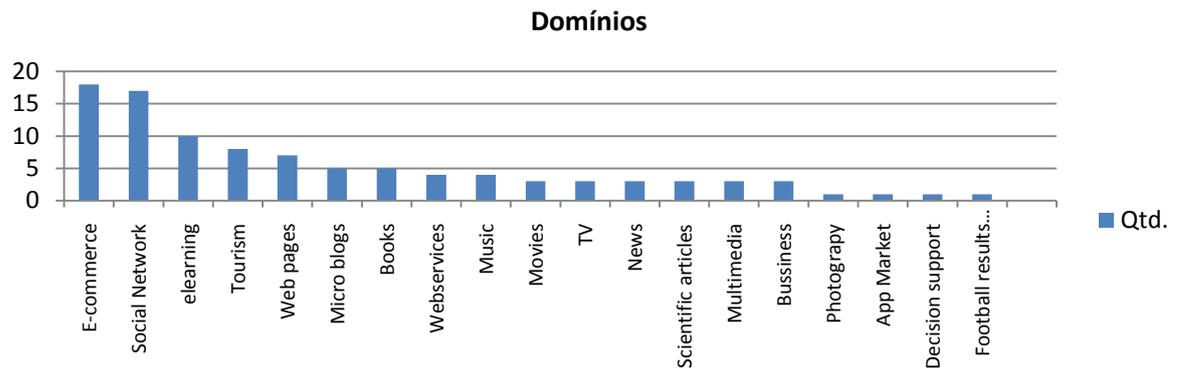


Gráfico 5 - Domínios de aplicação.

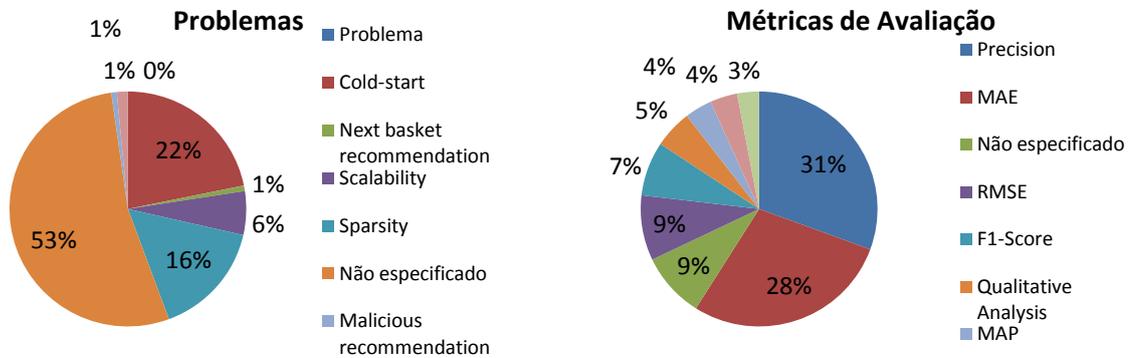


Gráfico 6 - Percentual de problemas recorrentes na área de sistemas de recomendação.

Gráfico 7 - Percentual das métricas de avaliação utilizadas nos trabalhos seleccionados.

3.6 TRABALHOS RELACIONADOS

Nesse capítulo são destacados dois trabalhos que propõem uma categorização ou agrupamento para sistemas de recomendação com base em uma revisão da literatura. Estes artigos se assemelham ao presente trabalho, pois realizam uma revisão em SR sem o foco em um determinado domínio de aplicação. Em (ADOMAVICIUS; TUZHILIN, 2005) é apresentada uma revisão da literatura sobre SR onde é apresentada uma categorização de SR com base no modo que a recomendação é feita. A categorização proposta por (ADOMAVICIUS; TUZHILIN, 2005) é utilizada como base no trabalho de (BOBADILLA et al., 2013) que considera que SR podem ser divididos em cinco grupos categorizados pelo método de filtragem, sendo elas a filtragem baseada em conteúdo, filtragem colaborativa, filtragem híbrida, filtragem demográfica e filtragem social.

Ambos os trabalhos apresentam alguns problemas conhecidos na área de SR como o problema da partida fria (*cold-start*), *overspecialition*, *sparsity problem*, entre outros. O trabalho de (BOBADILLA et al., 2013) também apresenta algumas métricas de avaliação utilizadas em SR como *MAE*, *precision* e *recall*.

Outra característica comum é que ambos apresentam uma visão do estado da arte em SR e indicam possíveis evoluções na área. Em comparação, o presente trabalho também apresenta uma visão do estado da arte em SR, porém a identificação do estado da arte foi usada como base para construir a abordagem central do trabalho, que é uma arquitetura aberta para construção e instanciação de SR independente do domínio de aplicação. A Tabela 4 apresenta de modo resumido uma comparação entre Ano, Metodologia e Proposta entre os trabalhos.

Trabalho	Ano	Metologia	Proposta
Toward the Next Generation of Recommender Systems: A Survey of the State-of-Art and Possible (ADOMAVICIUS; TUZHILIN, 2005)	2005	Não definido	Estado da arte, categorização, problemas e evoluções em SR.
Recommender Systems Survey (BOBADILLA et al., 2013)	2013	Pesquisa em trabalhos com base nas palavras-chaves mais relevantes em SR.	Estado da arte, categorização, problemas, métricas de avaliação e evoluções em SR.
<i>MMRecommender</i>	2017	Mapeamento Sistemático da Literatura	Estado da arte e arquitetura aberta MMRecommender.

Tabela 4 - Comparação entre os trabalhos relacionados.

Como resultado do mapeamento sistemático foi gerada uma tabela (Apendice A) contendo os dados extraídos dos artigos selecionados, desta forma os artigos foram categorizados quanto a proposta apresentada, sendo um modelo de recomendação ou um SR, método, técnica, domínio de aplicação, contexto (interno ou externo), *feedback* (implícito ou explícito), destinatário (usuário ou grupo), problema tratado e a métrica de avaliação utilizada. A partir da análise dos dados desta tabela foi gerada a primeira versão da arquitetura proposta, em seguida foi realizado um processo incremental de validação e contribuição do grupo de pesquisa evoluindo a arquitetura até a versão apresentada neste trabalho.

3.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Sistemas de recomendação são expansíveis e adaptáveis a domínios diferentes, sendo os mais comuns *e-commerce*, *e-learning*, turismo, ou domínios mais atuais como rede sociais e loja de aplicativos para dispositivos móveis (*App Market*). Alguns dos domínios identificados possuem um forte apelo econômico, sugerindo que o tema é relevante tanto para academia quanto para a indústria.

No mapeamento sistemático foram identificados 128 artigos que reúnem, categorizam ou descrevem temas relacionado à recomendação, categorizando esse tipo de sistema como adaptável e expansível a diferentes domínios. A quantidade crescente de publicações pode ser um indício de que ainda é um tema atual e de interesse na comunidade científica. Também foi possível perceber que aplicações em redes sociais ainda demandam muitas pesquisas na área.

Outros artigos sobre o tema foram identificados no mapeamento, entretanto, abordavam temas específicos (como partida fria) ou domínios restritos e foram descartadas por não oferecerem uma descrição expansível a múltiplos domínios.

Durante a extração de dados foi possível identificar técnicas distintas aplicadas no processo de recomendação, como processamento de linguagem natural, métodos heurísticos, algoritmos probabilísticos, métricas de similaridade, entre outras. Dessa forma é possível entender que contribuições em sistemas de recomendação possam surgir de áreas distintas da computação, como inteligência artificial, mineração de dados, engenharia de software, além de áreas com estudos multidisciplinares com a computação, como estatística e lingüística.

As características apresentadas podem contribuir para encontrar modelos, *frameworks* ou um consenso aplicado a SR independente do domínio de atuação. Essa

particularidade também foi encontrada durante a extração de dados quando foram identificados termos comuns na área que são usados de maneiras distintas ou sem uma definição formal.

Frente a esse cenário não foi possível responder, de forma direta, a questão de pesquisa primária, porém essa dificuldade serviu como indício de que um modelo genérico, independente de domínio, poderia ser útil para a academia e indústria. Como resultado do mapeamento e conhecimento prévio do grupo de pesquisa foi gerada uma arquitetura aberta de recomendação apresentada na seção posterior.

A questão de pesquisa secundária foi de extrema importância para coletar técnicas distintas utilizadas em SR e identificar diversas contribuições ao tema de áreas distintas. Dessa forma a pergunta secundária também auxiliou na composição dos itens presentes em etapas da arquitetura aberta.

O mapeamento sistemático apresentando foi importante para capturar características gerais da área, entender o estado da arte e prática ao analisar os métodos e técnicas aplicados, definir os problemas e métricas de avaliação mais recorrentes e compor uma arquitetura aberta de recomendação, expansível e adaptável a diversos domínios.

4. MMRECOMMENDER: PROPOSTA DE UMA ARQUITETURA ABERTA PARA SR

Este capítulo apresenta uma arquitetura aberta com os componentes necessários para a construção de SR, denominada MMRecommender (SIMOES et al., 2016) , identificados a partir dos 128 artigos selecionados no mapeamento sistemático e de modelos gerados pelo grupo de pesquisa.

A proposta define um processo composto de quatro etapas essenciais, e uma sub-etapa opcional, necessárias para instanciar ou representar modelos de recomendação: Extração, Filtragem, Método e Recomendação, além da sub-etapa Enriquecimento.

Segundo (PRESSMAN et al. 2016) o projeto de arquitetura representa a estrutura de dados e componentes necessários para construir o sistema computacional. A arquitetura dá uma visão geral da solução e sua representação explícita ajuda na tomada de decisão para o planejamento e implementação de sistemas de recomendação.

A proposta MMRecommender descreve como o sistema é estruturado e como seus componentes trabalham em conjunto. A arquitetura MMRecommender reflete o domínio e a lógica dos SR e retrata o fluxo lógico que se inicia com a definição dos usuários ou grupos de usuários, os dados de perfil e contexto, os algoritmos da abordagem de recomendação e a disponibilidade dos itens recomendados conforme exibido na Figura 3.

A partir do mapeamento sistemático e das propostas anteriores do grupo, notadamente na área de sistemas de recomendação de recursos educacionais (PEREIRA et al., 2014), (PEREIRA et al., 2015) e (ALMEIDA; CAMPOS; STROELE, 2015), os elementos dos componentes da arquitetura foram identificados e inseridos em uma tabela, auxiliando o processo incremental de desenvolvimento da arquitetura proposta no presente trabalho.

Para cada aplicação instanciada da arquitetura aberta MMRecommender é necessário definir as etapas Extração, Filtragem, Método e Recomendação e especificar os itens que as compõem. Desta forma é possível compor modelos de recomendação a partir da arquitetura aberta MMRecommender.

Extração: É a etapa onde as informações que irão compor o perfil do usuário ou do grupo serão extraídas, seja de forma implícita ou explícita. O contexto é o fator responsável pela forma de utilização do sistema e construção do perfil do usuário ou grupo. Foi utilizada a categorização apresentada por (FLEISHMANN; BASTOS; PERNAS, 2012) que divide o contexto em dimensões internas e externas.

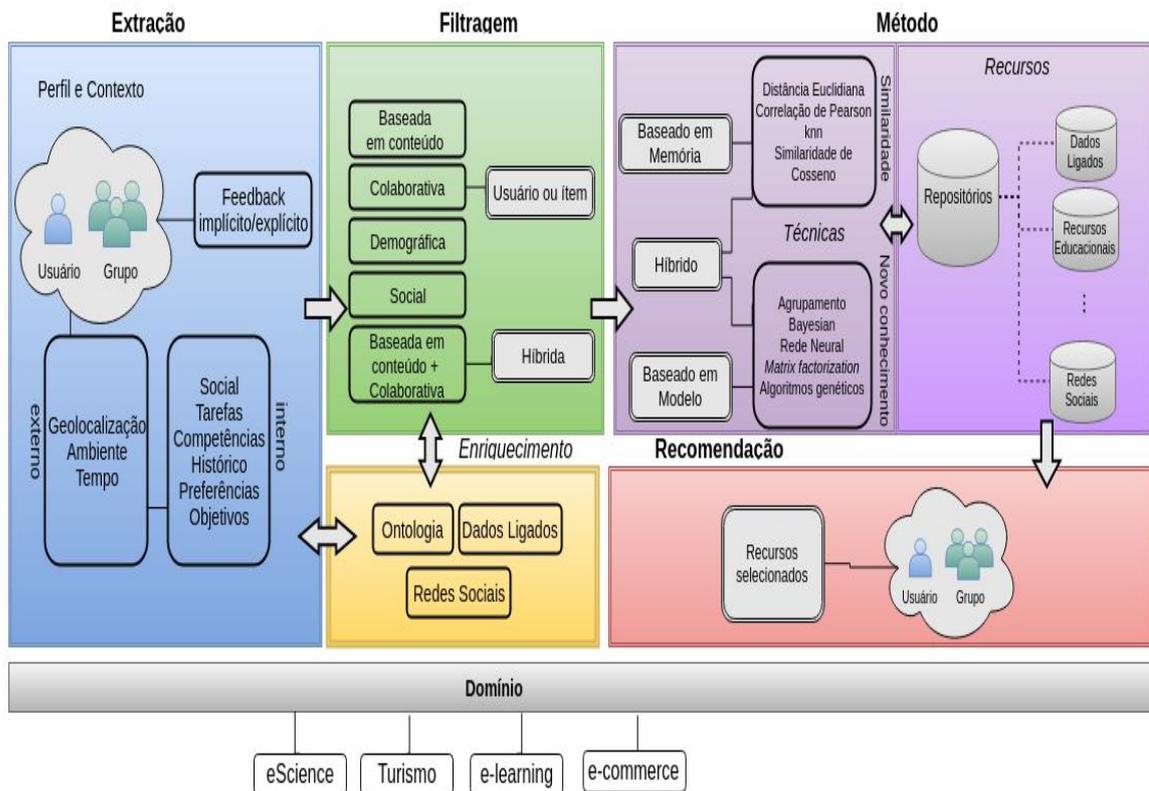


Figura 3 - Arquitetura conceitual MMRRecommender.

A sub-etapa **Enriquecimento** (Improvement) está ligada às etapas Extração e Filtragem, pois pode ser aplicada para o enriquecimento dos dados do usuário ou aos objetos que serão recomendados. Pode ser responsável pelo aprimoramento do perfil extraído através de informações contidas nas redes sociais, em dados ligados e ou em ontologias, permitindo assim adicionar informações complementares a base de dados de usuários ou objetos de recomendação. A utilização desta estratégia pode aumentar a assertividade da recomendação e mitigar o problema da partida fria. Esta é uma etapa opcional no modelo de recomendação.

Filtragem: Nessa etapa o algoritmo de filtragem é aplicado. Como dito anteriormente, segundo (BOBADILLA et al., 2013), uma categorização bem aceita na área divide os algoritmos em cinco tipos: Baseada em Conteúdo (*Content-based*) realiza as recomendações com base nas escolhas já feitas pelo usuário; Colaborativa (*Collaborative*) recomenda com base nas escolhas de outros usuários com perfil e preferências similares (existem dois subtipos *user-based* e *item-based*); Demográfica

(*Demographic*) que divide os usuários em grupos com preferências similares; Social (Social) baseado na recomendação em redes sociais e grupos de usuários, formada de modo explícito ou implícito e Híbrida (*Hybrid*) que é a combinação de duas ou mais técnicas.

Método: É a etapa em que a estratégia de recomendação é aplicada, sendo as mais comuns: *Model-based* (que cria um modelo que define o perfil e preferências do usuário ou grupos de usuários); *Memory-based* (também conhecido como *similarity-based*, podendo ser subdivido em *user-similarity* ou *item-similarity*) e *Hybrid* (que é a combinação das estratégias anteriores) (BOBADILLA et al., 2013). Os algoritmos de recomendação utilizam técnicas estatísticas, de mineração de dados, inteligência artificial, entre outras. A partir da aplicação do método os itens são recomendados.

A sub-etapa Recursos pode ser utilizada para fornecer recursos externos que serão recomendados, como por exemplo, vídeo aulas do *Youtube* (www.youtube.com) ou através de uma abordagem que utiliza dados ligados para representação de recursos educacionais (PEREIRA et al., 2015).

Recomendação: É a etapa final do processo de recomendação, onde os recursos são selecionados para serem apresentados aos usuários ou grupo de usuários.

Os SR podem adotar estratégias diferentes para apresentar as recomendações dependendo do domínio de aplicação: listas de itens, ordenação pela aderência ao perfil ou por avaliações de outros usuários, por associação entre preferências de usuários semelhantes ou por objetos recomendados com características similares.

Essa etapa está diretamente relacionada aos repositórios de recursos de onde sairão os itens que serão selecionados e apresentados aos usuários.

4.1 CONSIDERAÇÕES FINAIS DO CAPÍTULO

A arquitetura MMRecommender buscou contemplar os principais componentes de sistemas de recomendação, apresentada na forma de uma arquitetura aberta. Diferentes componentes de uma arquitetura definem se o sistema é aberto ou fechado como (ALSPAUGH et al., 2013): componentes de código fonte, componentes executáveis, serviços, interfaces e APIs, conectores, métodos de conexão e configurações. Nessa primeira etapa do projeto definimos um modelo arquitetural, que permite a configuração de sistemas de recomendação baseados em diferentes estratégias, domínios, restrições e interações.

As principais vantagens da proposta são:

- Capacidade de atender à flexibilidade exigida pelos requisitos dos diferentes domínios dos sistemas de recomendação;
- Disponibilidade de opções de modelos, técnicas e abordagens que permitam acesso a soluções já testadas em sistemas semelhantes;
- Contribuição para o desenvolvimento de sistemas de recomendação de alta qualidade e aderência para os usuários finais.

5. AVALIAÇÃO DA ARQUITETURA MMRECOMMENDER

Segundo WOHLIN et al (2012), dependendo do propósito da avaliação, se está se avaliando uma técnica, um método ou uma ferramenta, e dependendo das condições para uma investigação empírica, existem três principais tipos de estratégias de investigação: *surveys*, estudo de caso e experimento. Um *survey* é um sistema para coletar informações de ou sobre pessoas para descrever, comparar ou explicar seus conhecimentos, atitudes e comportamento. Já um estudo de caso no contexto de engenharia de *software* é uma investigação empírica, que se baseiam em diferentes fontes de evidências, usada quando o objeto de estudo é um fenômeno contemporâneo difícil de ser estudado de forma isolada. O experimento geralmente é realizado em laboratório e oferece maior nível de controle das variáveis envolvidas, manipulando uma ou algumas variáveis e mantendo outras fixas medindo o efeito do resultado. Segundo YIN (2001) o estudo de caso deve ser usado quando o pesquisador tem pouco controle sobre os acontecimentos, em contextos reais e sem controle total sobre as variáveis.

As principais características do estudo de caso são (RUNESON apud WOHLIN et al. 2012):

- É um tipo de estudo flexível, que lida com as características complexas e dinâmicas de fenômenos do mundo real;
- suas conclusões são baseadas em uma clara cadeia de provas, seja através de uma análise qualitativa ou quantitativa, coletadas de várias fontes diferentes, de forma planejada e consistente;
- o conhecimento existente pode ser baseado em uma teoria previamente estabelecida, ou através da construção de uma teoria.

Segundo (WOHLIN et al., 2012a) existem diferentes definições para estudos de casos em Engenharia de Software, é possível sintetizar conceitos propostos por diferentes autores em uma definição resumida de estudo de casos, sendo caracterizada pelo método experimental focado na investigação de fenômenos contemporâneos em um determinado contexto. Conforme os autores este método possui vantagens para a Engenharia de *Software*, entre elas é possível destacar: a flexibilidade para aplicar em cenários diversos e as conclusões são resultados através de diversas fontes de evidências, de natureza qualitativa ou quantitativa.

De acordo com WOHLIN et al (2012), a vantagem de um estudo de caso é que eles

são mais fáceis de planejar e mais realistas, mas a desvantagem é que os resultados dificilmente podem ser generalizados e são mais difíceis de interpretar.

Segundo WOHLIN et al (2012), o método de pesquisa é empírico quando um modelo é proposto e avaliado através de estudos empíricos, por exemplo, estudos de caso e experimentos. Segundo (DRESCH; LACERDA; ANTUNES, 2015), o estudo de caso é uma pesquisa empírica que busca melhor compreender um fenômeno contemporâneo, normalmente complexo no seu contexto real. São considerados valiosos, pois, permitem descrições detalhadas dos fenômenos normalmente baseados em fontes de dados diversas e asseguram que a investigação e o entendimento do problema sejam feitos em profundidade.

No caso desta pesquisa a melhor forma de avaliação são estudos de caso por se tratar de uma pesquisa empírica realizada em um contexto real, que deseja investigar a adequação da arquitetura para sistemas de recomendação que atenda a diferentes propostas e domínios.

A formalização dos Estudos de Caso será baseada em (DRESCH; LACERDA; ANTUNES, 2015), a fim de contribuir para a avaliação da questão de pesquisa formulada, e verificação dos artefatos desenvolvidos. Foram definidas as seguintes fases, adaptadas de estudos de caso WOHLIN et al (2012): (I) definição do estudo de caso; (II) formulação do objetivo; (III) planejamento; (IV) execução e observação das evidências; e (V) apresentação das evidências observadas.

Devido às características dos estudos de caso e a natureza adaptativa da arquitetura de sistemas de recomendação foram realizados três estudos de caso:

- Estudo de Caso I: BROAD-GRS aplicado ao domínio de *e-learning*;
- Estudo de Caso II: E-SECO pertinente ao domínio de *e-Science*;
- Estudo de Caso III: MKNOB no domínio turístico.

5.1 AVALIANDO A ARQUITETURA PARA UM SR EDUCACIONAL PARA GRUPOS

A recomendação em ambientes educacionais tem suas peculiaridades, pois os alunos possuem processos de aprendizagem próprios e alcançam diferentes níveis de competências (REZENDE et al., 2015).

Em ambientes sociais os SR são de particular importância, porque os usuários compartilham recursos formando grupos com interesses comuns (REZENDE et al., 2015)(RIBEIRO; FONSECA; FREITAS, 2013). Nesse cenário de interação um dos grandes

desafios desses sistemas é como lidar adequadamente com as preferências de cada integrante de um grupo para geração de uma recomendação conjunta (CARVALHO; MACEDO, 2014).

A opção por iniciar a avaliação da proposta de arquitetura apresentada nesta dissertação por um estudo de caso, em que a MMRecommender foi instanciada para um SR existente, no caso o BROAD-GRS (ALMEIDA; CAMPOS; STROELE, 2015), justifica-se pela necessidade de demonstrar a viabilidade do modelo proposto. Esse estudo contribuiu também para a avaliação preliminar da questão de pesquisa formulada na Introdução, a amenização de dificuldades técnicas pontuais da proposta e o refinamento do modelo.

Assim, esse estudo de caso I tem o objetivo avaliar a adequação do modelo arquitetural a um sistema de recomendação cuja arquitetura inicial não seguiu a proposta MMRecommender. Para tal foram instanciadas na arquitetura MMRecommender as etapas Extração, Filtragem, Método e Recomendação.

5.1.1 BROAD-GRS

Para avaliar a arquitetura no domínio educacional apresentamos o BROAD-GRS (ALMEIDA; CAMPOS; STROELE, 2015), um Sistema de Recomendação para Grupos, desenvolvido no contexto do projeto BROAD (Figura 4). A proposta desse sistema de recomendação evoluiu e inovou as propostas de (REZENDE et al., 2015) e (PEREIRA et al., 2014) com características identificadas em (ALMEIDA; CAMPOS; STROELE, 2015) e nos trabalhos relacionados da literatura (CASAGRANDE; KOZIMA; WILLRICH, 2015) e (LU et al., 2015).

Foi desenvolvido um protótipo capaz de extrair informações, definir o perfil educacional do grupo e recomendar recursos educacionais aos seus membros, utilizando a rede social *Facebook* (www.facebook.com).

A proposta adota uma Filtragem Híbrida, pois identifica o conteúdo relevante ao usuário, utilizando o seu perfil na formação do perfil do grupo (filtragem baseada em conteúdo) e recomenda com base na aderência ao perfil da maioria do grupo (filtragem colaborativa).

A recomendação é feita através de três abordagens: baseada em repositórios de recursos educacionais, em repositórios de Dados Ligados e em um repositório de vídeos. A recomendação dos recursos educacionais em repositórios se dá a partir da relação estabelecida entre as características do perfil do grupo e os metadados dos recursos educacionais. O projeto

BROAD já prevê a catalogação de recursos educacionais (NERY et al., 2012). Dentre as iniciativas de disponibilização de conteúdo através de Dados Ligados foram utilizadas neste trabalho a *DBpedia* (wiki.dbpedia.org/Datasets) e a *Open University* (data.open.ac.uk/). A recomendação em vídeos se dá através do *YouTube*.

5.1.2 Estudo de Caso I

O estudo de caso aqui descrito, avaliou a possibilidade de instanciar a arquitetura MMRecommender para um sistema de recomendação já desenvolvido e avaliado (ALMEIDA; CAMPOS; STROELE, 2015).

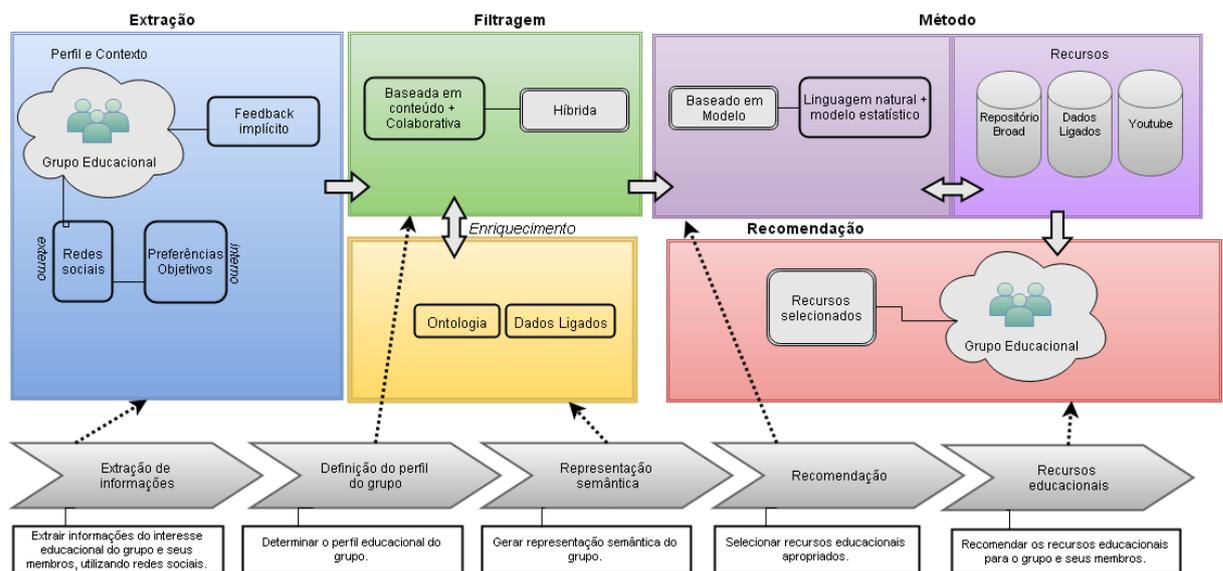


Figura 4 - Instância derivada do MMRecommender para o BROAD-GRS.



Figura 5 - Recomendação de recursos educacionais do repositório BROAD para o Grupo 1.

A arquitetura do BROAD-GRS é dividida em cinco módulos: (1) Camada de extração de informações; (2) Camada de definição do perfil do grupo; (3) Camada de representação semântica; (4) Camada de recomendação e (5) Camada de interface onde, os recursos são apresentados ao usuário. A Figura 4 apresenta o instanciamento da arquitetura MMRecommender para o BROAD-GRS.

A seguir descrevemos a avaliação da proposta BROAD-GRS (ALMEIDA; CAMPOS; STROELE, 2015) que analisou grupos de usuários na rede social *Facebook*. Foram criados grupos educacionais compostos de participantes voluntários sendo o Grupo 1, formado por 16 participantes, integrantes de uma Escola Técnica de Informática, e o Grupo 2 composto por 10 participantes de áreas distintas como Direito, Engenharia Ambiental e Ciência da Computação. Após cada membro permitir que o protótipo tivesse acesso às suas informações pessoais no Facebook, o perfil educacional do grupo foi definido. Com base nesse perfil, o tema educacional escolhido para o Grupo 1 foi “*Wireless Network*” enquanto o tema “Cotas Raciais” foi abordado no Grupo 2.

Na avaliação foram implementadas as três abordagens para os recursos a serem recomendados: repositório local de recursos educacionais BROAD, dados ligados e vídeos. Foi definida a estratégia de recomendar três recursos educacionais por abordagem. O protótipo indicou os recursos ao administrador do grupo, que repassou as informações para os membros do grupo. A Figura 5 apresenta os recursos educacionais, do repositório BROAD de objetos de aprendizagem, recomendados ao Grupo 1. No momento da avaliação, esse repositório possuía 74 recursos educacionais relacionados à área de Ciência da Computação.

Após a avaliação os participantes responderam um questionário contendo perguntas de múltipla escolha, usando a escala “Muito Ruim”, “Ruim”, “Razoável”, “Bom” e “Muito Bom”.

Os participantes do Grupo 1 avaliaram positivamente o processo de extração de seus dados, com 38% e 25%, respectivamente, de avaliações “Bom” e “Muito Bom”, para o Grupo 2 as avaliações positivas foram de 30% e 40%, para “Bom” e “Muito Bom” respectivamente.

Os recursos recomendados para o Grupo 1 obtiveram 69% de avaliações positivas, enquanto o Grupo 2 obteve 90% de avaliações positivas. A avaliação da recomendação baseada em Dados Ligados obteve 63% de satisfação, considerando as avaliações “Bom” e “Muito Bom” para o Grupo 1 e 80% de avaliações “Bom” e “Muito Bom” para o Grupo 2. Nas recomendações em vídeos, os dados são ainda melhores, 86% para o Grupo 1 e 90% para o Grupo 2, de satisfação considerando as avaliações “Bom” e “Muito Bom”. Os dados

coletados sugerem que a proposta BROAD-GRS foi eficaz na recomendação de recursos educacionais no contexto dos participantes do Grupo 1 e Grupo 2, além de apresentar evidências que os recursos educacionais em vídeo são mais aderentes ao perfil dos grupos analisados.

O protótipo utiliza como base dados de perfil do *Facebook* para a formação de grupos, dessa forma dados privados do usuário podem dificultar a identificação precisa de interesses do grupo. A falta de informações relevantes para a formação de grupos também é um ponto importante que pode dificultar a recomendação visto que o *Facebook* é uma rede social que não possui o foco na formação de grupos educacionais.

A avaliação não considerou aspectos temporais como a variação do interesse e características dos participantes do grupo, tais aspectos serão tratados em trabalhos futuros.

O Gráfico 8 apresenta o quantitativo de respostas dos usuários a respeito da aceitação em relação às suas características pessoais extraídas, bem como as características de cada grupo respectivamente. O Gráfico 9 apresenta a aceitação dos usuários em relação às recomendações educacionais feitas ao grupo respectivamente, levando em consideração suas características individuais e do grupo.

Os participantes do Grupo 1 avaliaram positivamente o processo de extração de seus dados, com 38% e 25%, respectivamente, de avaliações “Bom” e “Muito Bom”, para o Grupo 2 as avaliações positivas foram de 30% e 40%, para “Bom” e “Muito Bom” respectivamente.

Os recursos recomendados para o Grupo 1 obtiveram 69% de avaliações positivas, enquanto o Grupo 2 obteve 90% de avaliações positivas. A avaliação da recomendação baseada em Dados Ligados obteve 63% de satisfação, considerando as avaliações “Bom” e “Muito Bom” para o Grupo 1 e 80% de avaliações “Bom” e “Muito Bom” para o Grupo 2. Nas recomendações em vídeos, os dados são ainda melhores, 86% para o Grupo 1 e 90% para o Grupo 2, de satisfação considerando as avaliações “Bom” e “Muito Bom”. Os dados coletados sugerem que a proposta BROAD-GRS foi eficaz na recomendação de recursos educacionais no contexto dos participantes do Grupo 1 e Grupo 2, além de apresentar evidências que os recursos educacionais em vídeo são mais aderentes ao perfil dos grupos analisados.

O protótipo utiliza como base dados de perfil do *Facebook* para a formação de grupos, dessa forma dados privados do usuário podem dificultar a identificação precisa de interesses do grupo. A falta de informações relevantes para a formação de grupos também é

um ponto importante que pode dificultar a recomendação visto que o *Facebook* é uma rede social que não possui o foco na formação de grupos educacionais.

A avaliação não considerou aspectos temporais como a variação do interesse e características dos participantes do grupo, tais aspectos serão tratados em trabalhos futuros.

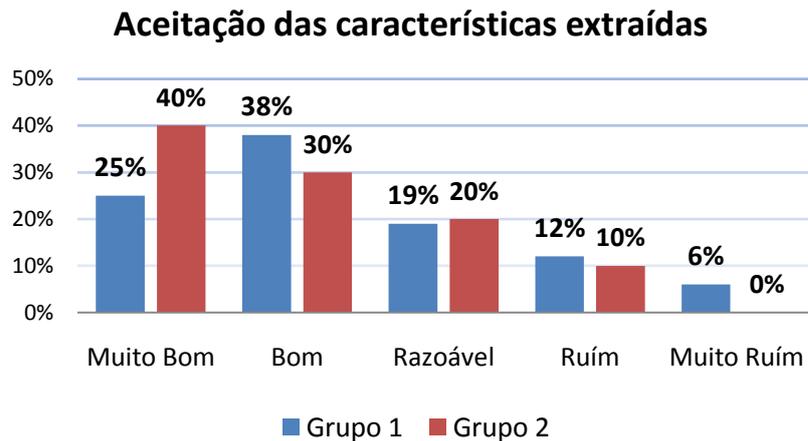


Gráfico 8 - Aceitação das características extraídas dos Grupos.

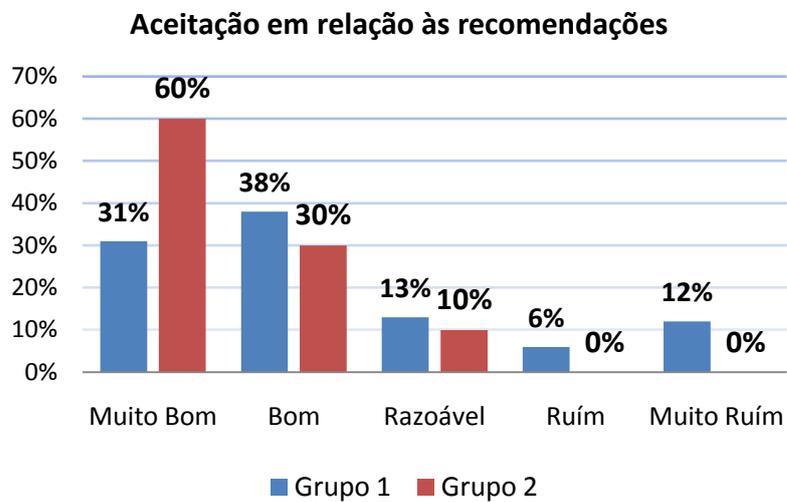


Gráfico 9 - Aceitação dos usuários em relação às recomendações feitas ao Grupo 1.

5.1.3 Evidências Observadas

O BROAD-GRS (ALMEIDA; CAMPOS; STROELE, 2015), um SR para grupos educacionais em redes sociais foi desenvolvido e avaliado. A recomendação utilizando três abordagens (repositório de dados, dados ligados e vídeos) foi considerada satisfatória e complementar. O uso de dados ligados e de um repositório de vídeos no processo de seleção de recursos permitiu ampliar as possibilidades de recomendações oferecidas aos usuários, não ficando restrito a repositórios de recursos educacionais com temas específicos. Além disso, o envio dos recursos educacionais através da recomendação em redes sociais oferece ao usuário a utilização dos ícones de interatividade do ambiente, que permitem curtir e compartilhar suas experiências sobre um recurso educacional recomendado.

As evidências observadas neste estudo de caso e os resultados do mapeamento sistemático foram publicados em (SIMOES et al., 2016) evidenciando que a instanciação da arquitetura permitiu manter as características originais do BROAD-GRS (ALMEIDA; CAMPOS; STROELE, 2015), um SR para grupos educacionais em redes sociais, sugerindo que a arquitetura aberta pode ser aplicada em sistemas de recomendação existentes, provendo uma forma simplificada de visualização das etapas e componentes do sistema.

Este estudo de caso também contribuiu para a realização do trabalho (ABDALLA et al., 2017) que apresentou evoluções da proposta apresentada anteriormente no estudo de caso.

5.2 AVALIANDO A ARQUITETURA PARA UM ECOSISTEMA DE SOFTWARE

Workflow científico é um recurso utilizado no domínio de *e-Science* e está relacionado à divisão em processos sequenciais de um experimento científico. Geralmente para a implementação de um *workflow* são necessários conhecimentos distintos em computação e no domínio de atuação do *workflow*. Não raramente um experimento científico pode envolver diversas áreas do conhecimento, por isso a concepção de experimentos científicos não é uma tarefa trivial.

Nesse contexto surgiram comunidades virtuais científicas que pretendem diminuir o esforço de desenvolvimento de *workflows* e aumentar os índices de reuso de *workflows* e colaboração entre grupos de pesquisa. Nessas comunidades cientistas podem disponibilizar

publicamente *workflows* através da *Internet*.

Workflows disponíveis na *Internet* são geralmente implementados na forma de *web-service*, que é uma forma de serviço computacional muito utilizado em sistemas de informação. Características comuns a *web-services* também estão presentes em *workflows* científicos, como requisições utilizando o protocolo HTTP (*Hypertext Transfer Protocol*) e retorno em formatos JSON (*Javascript Object Notation*) e XML (*Extensible Markup Language*). Seguindo o conceito de serviço o processamento e implementação fica abstraído para o usuário (TIAN et al., 2014b), sendo necessário apenas conhecer os dados de entrada e saída do serviço.

Esta seção aborda detalhes da arquitetura de um ecossistema colaborativo de software científico onde um modelo de recomendação derivado da arquitetura MMRecommender foi planejado e implementado.

O objetivo desse estudo de caso II é instanciar a arquitetura MMRecommender, como módulo de um sistema que já permitia analisar o modelo de recomendação com o propósito de oferecer recomendações de serviços mais aderentes ao perfil do usuário, com relação aos serviços utilizados com mais frequência e recentemente utilizados, do ponto de vista do usuário do E-SECO no contexto de *e-Science*. Para tal foi planejado e implementado o módulo de recomendação e avaliado com dados reais do ecossistema e de redes sociais científicas.

5.2.1 Ecossistema E-SECO

O sistema de recomendação apresentado nesta seção foi desenvolvido e aplicado ao domínio de *e-Science* utilizando o ecossistema de *software* científico e colaborativo chamado E-SECO (FREITAS et al., 2015). Este ecossistema pode realizar a prototipação de experimentos, que é a definição de um fluxo de trabalho para cada etapa necessária no experimento utilizando serviços externos provenientes de sites como *myExperiment* (www.myexperiment.org) e *BioCatalog* (www.biocatalog.org). O E-SECO (Figura 6) oferece uma interface de busca e visualização de serviços e *workflows* externos para a composição do experimento, apresentando uma lista ordenada dos serviços de acordo com a comparação sintática do termo buscado e o nome do serviço.

O E-SECO possui uma arquitetura complexa que implementa o conceito de Linha de Produto de *Software* (LPS) ao trabalhar com módulos separados que podem ser combinados

na composição final do *software*. Também possui elementos de colaboração entre desenvolvedores, usuários e grupos de pesquisa. Seus principais módulos são:

- *Development Environment*: Responsável pelo gerenciamento e versionamento do repositório de código fonte, notificações de erros e envio de novas *features* através de *pull-requests*. Através desse ambiente é possível que desenvolvedores contribuam com a evolução do ecossistema implementando novas *features* ou artefatos que podem ser integrados ao E-SECO.
- *Execution Environment*: É o ambiente responsável pela execução dos serviços utilizados na prototipação dos experimentos.
- *External Platforms Third-Party Applications*: É a camada responsável por consultar APIs externas ao E-SECO, como por exemplo, serviços do *myExperiment* e *BioCatalogue*.

O modelo de recomendação proposto neste trabalho utiliza, principalmente, os módulos *Development* e *External Platforms Third-party*.

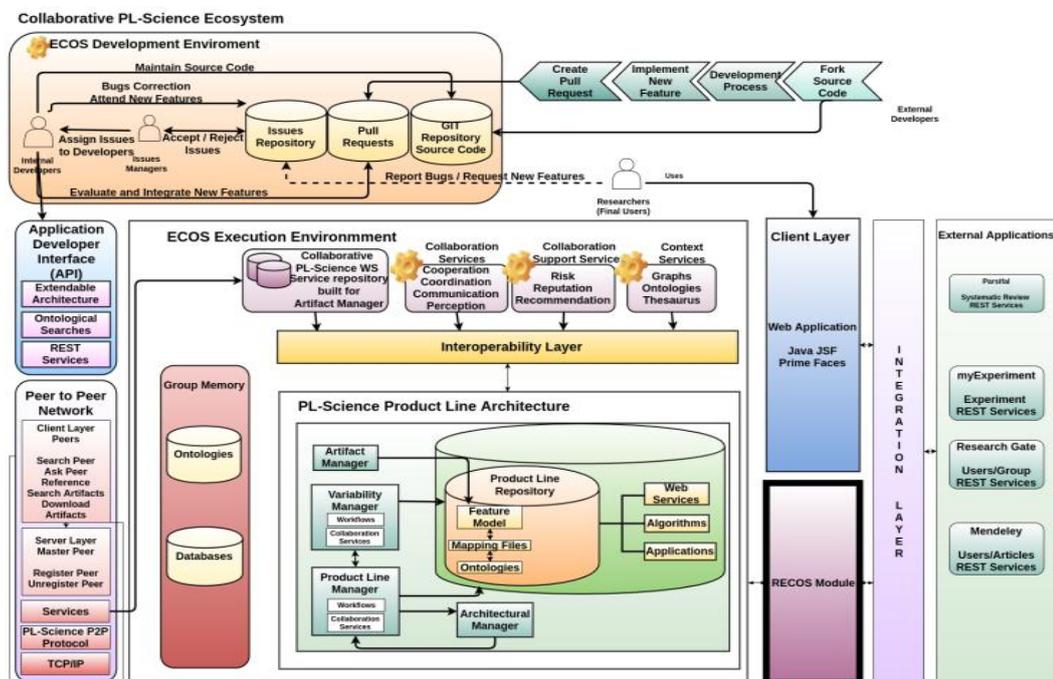


Figura 6 - Arquitetura do E-SECO.

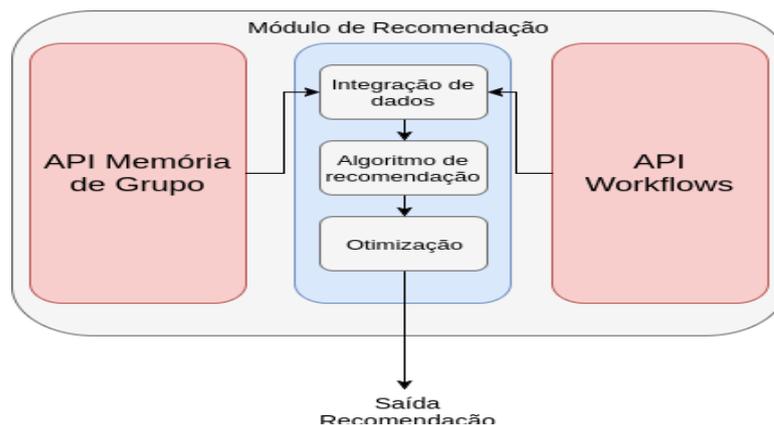


Figura 7- Módulo de Recomendação do E-SECO.

5.2.2 Estudo de Caso II

O E-SECO pode ser caracterizado com um sistema que utiliza a recomendação como meio, e não como fim, ou seja, seu objetivo principal não é recomendar serviços científicos, é auxiliar pesquisadores na etapa de prototipação de experimentos, além de prover ferramentas que contribuem para a comunicação entre pesquisadores geograficamente distribuídos.

Nesse contexto o segundo estudo de caso visa utilizar a arquitetura MMRecommender como um módulo de recomendação para serviços científicos que podem ser utilizados durante o processo de prototipação.

O modelo apresentado atua durante a busca de serviços externos ao E-SECO, desta forma o algoritmo reordena a lista resultante da busca com base em fatores inferidos de *rating* e *time*, que são calculados durante o processo de recomendação para cada serviço utilizado no sistema, logo, a recomendação pode ser definida como baseada no conteúdo (BOBADILLA et al., 2013).

O módulo de recomendação é implementado como um componente que pode ser integrado ao sistema e processa dados de entrada e saída através de *APIs*. Esta característica é importante para que o módulo de recomendação possa ser acoplado, ou não, ao sistema, mantendo as características de LPS.

A Figura 7 mostra como o módulo de recomendação é integrado no E-SECO, onde é possível perceber que os dados de entrada são provenientes da *API* de memória de grupo, que coleta os dados de utilização (*feedback* implícito) do sistema e também da *API* de comunicação com serviços externos ao E-SECO.

Como *workflows* científicos são fornecidos na forma de *web-services* o termo serviço será utilizado para serviços e *workflows*.

5.2.2.1 Identificação de Parâmetros para Recomendação de Serviços Científicos

Para identificar quais fatores podem ser considerados importantes na recomendação de serviços científicos foi criado um *dataset* com dois mil serviços originados do repositório *myExperiment*. Em seguida foram analisadas as relações entre o tempo de vida e a quantidade de *downloads*, a fim de identificar uma possível relação entre o tempo de existência dos serviços e a quantidade de *downloads*.

A quantidade de *downloads* do serviço foi analisada, pois é possível inferir que em um repositório com muitos serviços, sendo alguns com a mesma finalidade, o serviço com uma quantidade maior de *downloads* é o serviço mais popular da comunidade, essa informação pode ser considerada como uma métrica de qualidade do serviço.

5.2.2.2 Tempo de vida

Para o cálculo do parâmetro tempo de vida foi considerada a diferença, em anos, entre a data de análise e a data de publicação no repositório, em seguida foi calculada a média do tempo de vida de todos os serviços disponíveis nos *datasets*. O valor resultante foi 5,8 anos. Considerando esse valor os serviços foram ordenados de forma decrescente, de acordo com a quantidade de *downloads*. Em seguida os duzentos primeiros (10% da amostra) foram avaliados.

Serviços com um tempo de vida maior que a média foram considerados positivos (Bom), e abaixo negativos. Os resultados da análise da amostra são apresentados na

Tabela 5.

Dados	
Média vida (anos)	5,78
Máx. tempo de vida (anos)	8,55
Mín. tempo de vida (anos)	3,75
Desvio Padrão (anos)	1,41
Qtd. "Bom" Tempo de vida (anos)	190

Tabela 5 - Comparação do tempo de vida.

Para alcançar uma granularidade maior na avaliação dos serviços foi criada uma escala de pontuação com base no desvio padrão do tempo de vida dos serviços presentes na amostra. A Tabela 6 apresenta essa escala criada, assim como o agrupamento dos serviços

presentes em cada grupo de pontuação.

Escala	Pontuação	Qtd.
+2 Desvio Padrão	3	163
+1 Desvio Padrão	2	27
Média	1	8
-1 Desvio Padrão	0	2

Tabela 6 - Escala de pontuação com base no desvio padrão.

É necessário considerar que, quanto maior o tempo de vida do serviço, maior o tempo em que ele está disponível no repositório, conseqüentemente maior pode ser a quantidade de downloads, por isso as avaliações explícitas dos usuários, que são valores inteiros entre 0 e 5, foram estudadas.

Entretanto, a análise das avaliações explícitas apresentou três dificuldades, sendo a primeira a baixa quantidade de avaliações, dos 2000 *workflows* do *dataset* apenas 100 foram avaliados, com um total de 128 avaliações. A segunda dificuldade está na subjetividade das avaliações numéricas, ou seja, valores iguais podem ter pesos diferentes para cada usuário. E por último, as avaliações não estão relacionadas com a versão do serviço, ou seja, não considera as atualizações e possíveis melhorias.

A partir dos dados apresentados na

Tabela 5 e Tabela 6 é possível inferir que o tempo de vida pode ser usado como um fator positivo de qualidade de serviços no repositório myExperient. Devido às particularidades nos dados de avaliações explícitas elas foram desconsideradas para a construção da proposta, permanecendo apenas a análise com relação ao tempo de vida e a aceitação da suposição.

5.2.2.3 Fatores de Similaridade

O fator *rating* (Equação 1) é uma propriedade presente em todos os serviços utilizados no E-SECO e mede, de forma implícita, a avaliação do usuário àquele serviço (S). O *rating* é definido pela razão entre a frequência que um serviço (Si) foi utilizado, e o total de serviços utilizados. A alteração do viés do usuário é trabalhada através deste fator, por exemplo, caso um serviço muito utilizado seja substituído por outro, o fator *rating* será menor, logo, poderá ser menos recomendado com o tempo.

$$rating = \frac{freq(s_i)}{total(s)} \quad (1)$$

Time é um fator que mede características de todos os serviços, como data de criação, atualização e a data mais recente em que o serviço foi incluído em um experimento no E-SECO. Para o cálculo deste fator a data atual, ou seja, a data em que a recomendação será realizada é utilizada como parâmetro. O fator *time* é definido pela soma dos fatores: *fresh* e *bias*, que medem a atualização e o viés de preferência, respectivamente mostrados nas equações 2 e 3.

O fator *fresh* pode ser considerado um fator de qualidade do serviço, já que serviços que são frequentemente atualizados podem implicar em um nível de qualidade superior em relação a serviços desatualizados (TIAN et al., 2014b).

O fator *bias* mede a preferência atual dos usuários em relação à escolha de serviços mais ou menos recentes, logo, se em um determinado período os usuários optarem por utilizar serviços mais novos a recomendação considera essa preferência atual, considerando a alteração do viés de preferências.

$$fresh = \frac{(today-createdAt) + (today-updatedAt)}{today-createdAt} \quad (2)$$

$$bias = \frac{latestUsed-createdAt}{today-createdAt} \quad (3)$$

5.2.2.4 Modelo de Recomendação

O modelo de recomendação inicia-se a partir da busca de serviços através do E-SECO por um usuário. Nesse momento são calculados os fatores de todos os serviços utilizados e colocados no plano cartesiano com coordenadas que representam *rating* (ordenadas) e *time* (abscissas) para cada eixo. A Figura 8 mostra um exemplo de serviços no plano cartesiano.

Em sequência é calculado o valor médio dos serviços através da média geométrica, que equivale à tendência central entre um conjunto de serviços (Figura 9). Este ponto é usado como um modelo de preferência ou tendência de escolha do usuário.

Por fim é traçado um novo plano cartesiano com origem no ponto médio, então os serviços provenientes da busca do usuário recebem valores de *rating* e *time* e são inseridos no plano cartesiano (*rating* x *time*). A Figura 10 mostra a divisão em quatro quadrantes com origem no ponto médio e os serviços pesquisados pelo usuário.

A distância euclidiana pode ser usada como cálculo de similaridade (BOBADILLA et al., 2013), sendo que a distância entre o modelo de preferência e os serviços indicam o grau de similaridade, ou seja, quanto menor a distância maior a similaridade entre o serviço e a preferência do usuário.

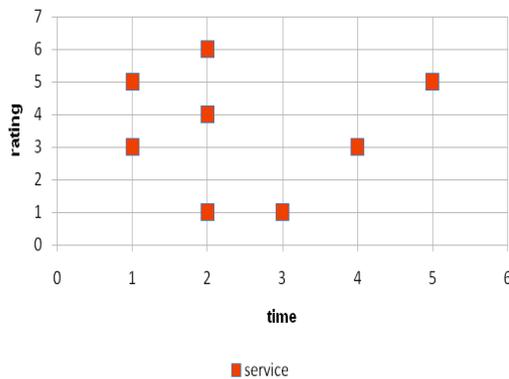


Figura 8 - Serviços utilizados no E-SECO inseridos no plano cartesiano *rating* e *time*.

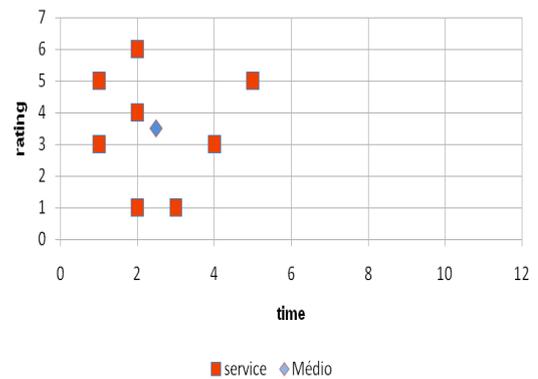


Figura 9 - Ponto geométrico dos serviços utilizados no E-SECO.

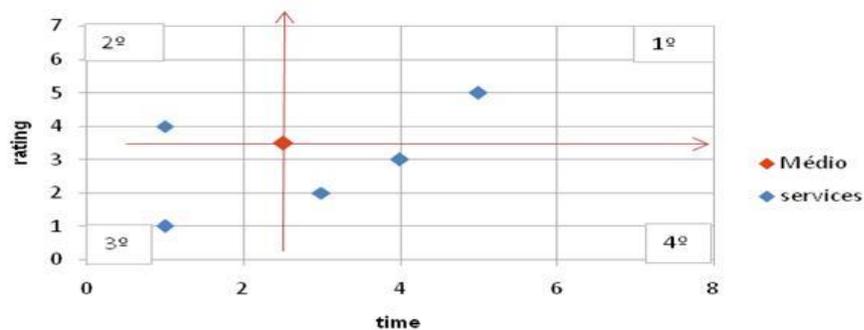


Figura 10 - Novo plano cartesiano com a média geométrica e os serviços retornados na busca.

Porém, considerar apenas a distância euclidiana pode resultar em recomendações não aderentes ao perfil do usuário, como por exemplo, os serviços localizados no terceiro quadrante do plano cartesiano. Estes serviços possuem valores de *rating* e *time* menores que o ponto médio (preferência do usuário), por outro lado, os serviços localizados no primeiro quadrante possuem valores maiores que o ponto médio. Sobre esse ponto de vista os quadrantes são priorizados e, em seguida, cada serviço localizado no quadrante é ordenado de acordo com a distância euclidiana entre o serviço e o ponto médio.

O fluxo completo para realizar a recomendação é apresentado na Figura 11.

A partir da arquitetura aberta proposta foi instanciada a arquitetura do modelo de recomendação aplicado ao E-SECO (Figura 6) com as etapas de:

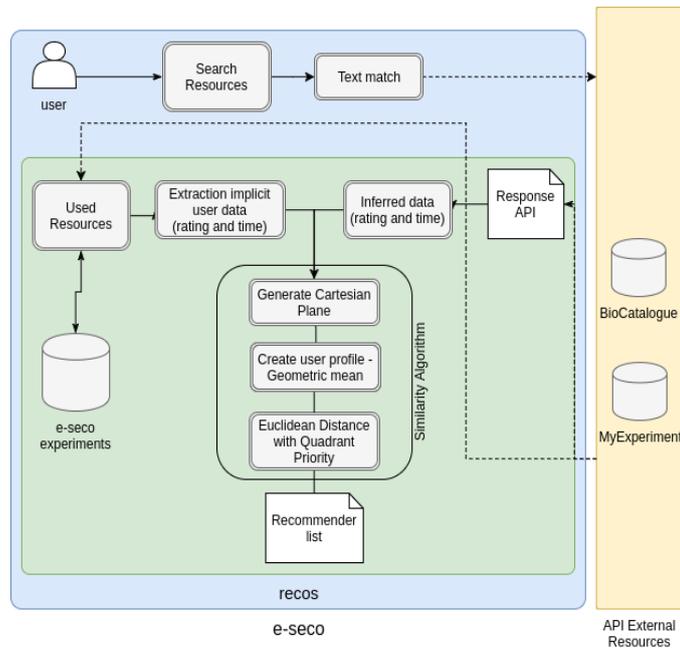


Figura 11 - Fluxo de recomendação do E-SECO.

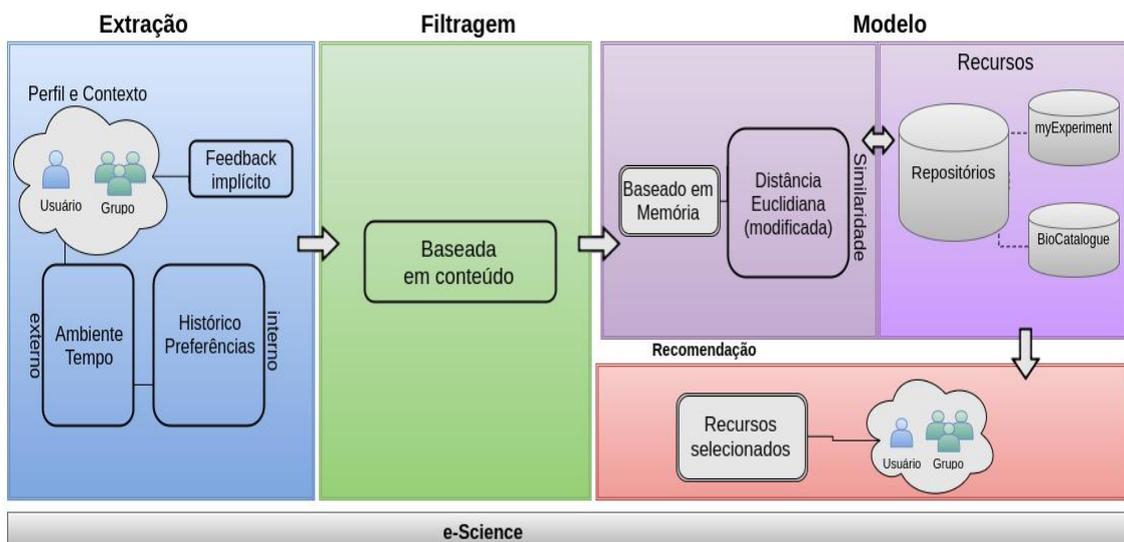


Figura 12 - Arquitetura de recomendação do E-SECO.

- **Extração:** Para gerar o perfil implícito os dados de prototipação dos experimentos, como serviços utilizados, data de uso e criação, são extraídos do banco de dados do E-SECO que utiliza o conceito de memória de grupo.
- **Filtragem:** Para evitar o problema da partida-fria presente na filtragem colaborativa o modelo utiliza a filtragem baseada em conteúdo onde os valores de rating e time são gerados para cada serviço utilizado.
- **Modelo:** Responsável por inferir os valores dos serviços retornados na busca do usuário, efetuar o cálculo de similaridade e reordenar a lista de serviços. O

método de cálculo da distância euclidiana modificado foi escolhido porque é um método que pode facilmente ser adaptado a qualquer contexto e possui baixa complexidade, resultando em um método eficiente que não compromete o desempenho do software ao efetuar o cálculo de recomendação.

- **Recomendação:** É o processo final onde os serviços são apresentados. No E-SECO todos os serviços retornados dos repositórios são ordenados de forma decrescente de similaridade, ou seja, do serviço mais recomendado ao menos recomendado.

5.2.3 Avaliação

Para realizar o estudo de caso foi criado um novo usuário e um experimento no E-SECO e em seguida alguns serviços do *myExperiment* foram adicionados na etapa de prototipação. Os serviços adicionados foram escolhidos de forma aleatória utilizando como base a palavra ‘*protein*’ e foram identificados apenas pelo *ID* presente no *myExperiment*. Os serviços escolhidos foram: 310, 477 e 703, entre os 200 serviços com maior quantidade de *downloads* da amostra.

Após a conclusão da prototipação foi realizada uma busca pelo mesmo termo utilizado na escolha aleatória dos serviços, porém com duas variações no ambiente. Na primeira variação o modelo de recomendação estava desativado, retornando a lista de serviços igual ao *myExperiment*. Na segunda variação foi ativado o modelo de recomendação, que realizou uma reordenação na lista de acordo com o perfil do usuário do estudo de caso.

A Tabela 7 apresenta, de modo simplificado, os dez primeiros serviços da lista de recomendação, o ID de cada serviço, sua data de criação, posição na lista e os valores de *rating* e a posição na lista sem recomendação, resultante do primeiro ambiente.

Através da tabela apresentada é possível perceber que o modelo de recomendação priorizou serviços com tempo de vida maiores, como os serviços 71, 115 e 124, que subiram mais de 15 posições na lista mesmo com valores de *rating* iguais a zero. O modelo também manteve nas posições iniciais os serviços outrora utilizados, com valores de *rating* maiores. Esse comportamento está aderente à proposta apresentada, onde os serviços que já foram utilizados e os serviços com mais tempo de vida devem ser priorizados.

Com Recomendação				
Id	Create	Rating	Posição	Posição s/ rec.
310	2008-07-12 10	4	1	1
703	2015-11-16 17	5	2	3
213	2008-06-01 12	1	3	6
71	2007-11-09 10	0	4	18
115	2007-12-10 22	0	5	19
124	2008-01-09 12	0	6	20
368	2008-07-30 16	0	7	8
216	2008-10-26 21	0	8	7
152	2008-11-27 10	0	9	4
160	2008-12-07 20	0	10	5

Tabela 7 - Comparação das listas geradas com e sem a recomendação

5.2.4 Evidências Observadas

No Estudo de Caso II, o desenvolvimento do módulo de recomendação do E-SECO a partir da arquitetura proposta sugere que o MMRecommender pode auxiliar na concepção de uma arquitetura aderente ao contexto específico e ao sistema já existente. O estudo de caso realizado apresenta evidências de que o modelo de recomendação implementado no E-SECO é eficiente, porém, é ainda necessário realizar experimentos para avaliar se a recomendação pode auxiliar no processo de prototipação de experimentos.

Durante as etapas de prototipação e avaliação da recomendação os serviços não foram avaliados de acordo com a sua funcionalidade, devido à falta de um especialista na área de biologia. Dessa forma foi possível avaliar apenas pelos fatores de *rating* e *time*.

Durante a análise do tempo de vida dos serviços também não foi possível analisar o tempo e quantidade de atualizações. Esta informação poderia sugerir outros aspectos importantes na recomendação dos serviços, porém estes dados não são divulgados pelo *myExperiment* e *BioCatalog*.

Os resultados deste estudo de caso foram publicados em (SIMOES et al., 2017) evidenciando que é possível utilizar as características de colaboração do E-SECO oferecendo uma forma de compartilhamento de serviços entre grupos de pesquisa. Também é possível trabalhar temas inerentes a sistemas de recomendação como o tratamento da partida-fria utilizando um processo de enriquecimento de perfil através de redes sociais científicas.

5.3 AVALIANDO A ARQUITETURA PARA UM SISTEMA DE RECOMENDAÇÃO TURÍSTICO

O Mknob, *Multilingual Knowledge Base* (Base de conhecimentos Multilíngüe) (<http://www.ufjf.br/framenetbr/m-knob/>) é um aplicativo de guia turístico personalizado, capaz de extrair informações do usuário e recomendar atrações turísticas e eventos esportivos alinhados com o perfil e contexto do usuário. Fruto de um trabalho interdisciplinar entre o grupo de pesquisa FrameNet Brasil (<http://www.ufjf.br/framenetbr>), ligado ao Departamento de Linguística, e ao NEnC (<http://www.ufjf.br/nenc/>) (Núcleo de Engenharia do Conhecimento) ligado ao Departamento de Ciência da Computação, ambos da UFJF.

Lançado durante o evento olímpico RIO 2016 na forma de *web-app* responsivo, ou seja, pode ser acessado através de dispositivos móveis ou computadores. O aplicativo pode efetuar recomendações com base no perfil implícito e explícito do usuário. Os dados de perfil implícitos são coletados através da rede social *Facebook*, durante o processo de *login* no aplicativo. A qualquer momento o usuário também pode completar o seu cadastro informando quais são seus interesses, dessa forma o perfil explícito é armazenado.

O usuário do Mknob pode escolher entre dois tipos de recomendação, sendo o primeiro a recomendação de pontos turísticos ou eventos esportivos com base no perfil do usuário, e o segundo com base na localização geográfica e perfil, filtrando os locais com base no perfil e priorizando os locais mais próximos. A Figura 13 mostra uma lista de locais recomendados com base no perfil do usuário e os detalhes do local selecionado.

O aplicativo também possui características de gamificação, além de um dicionário multilíngüe com termos esportivos e turísticos, porém esse trabalho irá abordar apenas as características de recomendação.

O Mknob tem como objetivo recomendar locais turísticos e atrações esportivas, além de oferecer um dicionário multi-idiomas, por isso pode ser categorizado como um sistema que utiliza a recomendação como fim.

O objetivo desse estudo de caso III é instanciar a arquitetura MMRecommender, para auxiliar e guiar o processo de definição e desenvolvimento do modelo de recomendação utilizado no Mknob. Neste contexto, a arquitetura MMRecommender foi utilizada para auxiliar a criação do modelo de recomendação utilizado no Mknob, com as etapas e itens e no processo de validação com pesquisadores da área. Nas etapas posteriores de implementação e testes o modelo de arquitetura gerado serviu como documentação para a

avaliação do modelo de recomendação implementado. Desta forma, é possível destacar que a arquitetura auxiliou em três etapas do processo de construção do modelo de recomendação, sendo eles:

- Definição e construção do modelo de recomendação.
- Apresentação de um modelo visual de fácil entendimento para os interessados no projeto, melhorando a comunicação.
- Fonte de documentação durante o processo de implementação e testes.

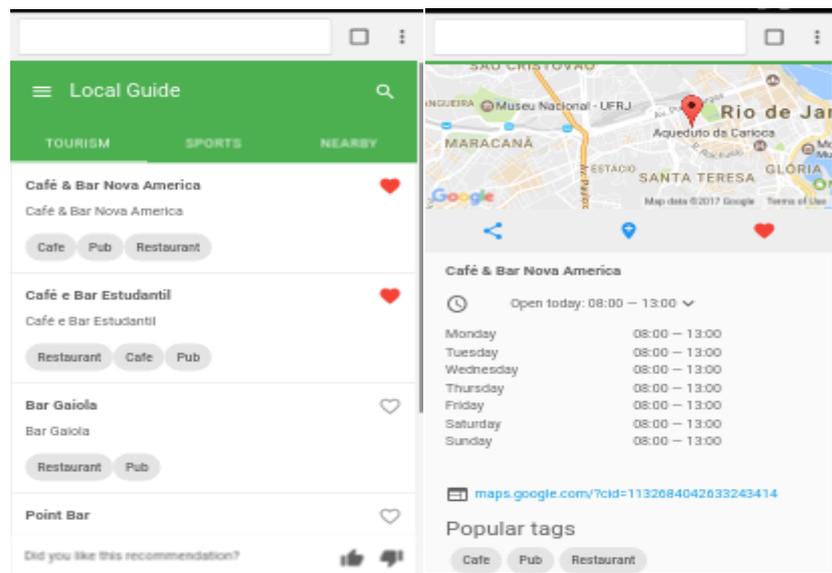


Figura 13 - Tela com recomendação de locais turísticos e detalhes do local.

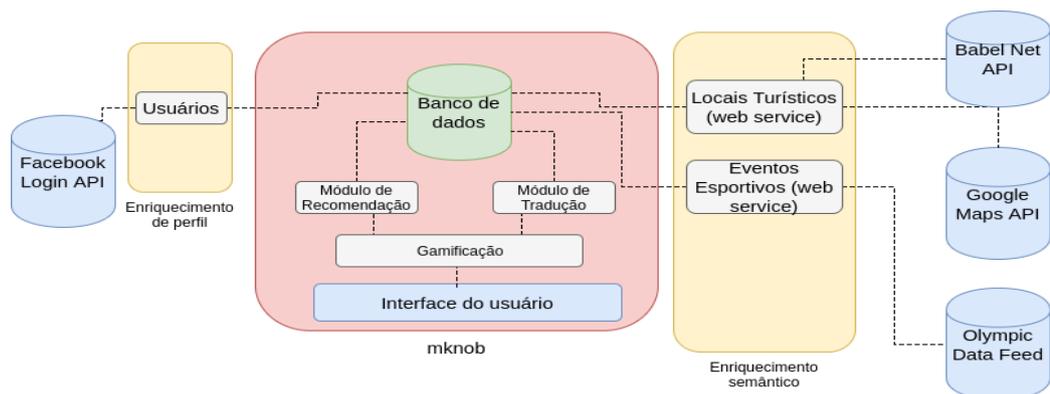


Figura 14 – Arquitetura do Mknob.

5.3.1 Arquitetura do Sistema MKNOB-RECOMMENDER

O modelo de recomendação do aplicativo Mknob mede, resumidamente, o nível de aderência do perfil e contexto do usuário com características dos locais turísticos e eventos esportivos (objetos de recomendação). Portanto, foi necessário criar um repositório com informações dos usuários e objetos recomendados. Nessa etapa foram desenvolvidos *web-services*, para as consultas na base de dados, e *APIs (application program interface)* externas ao Mknob, para identificar os pontos turísticos e os dados e a localização dos usuários.

A base de dados *Olympic Data Feed* (odf.olympictech.org/), disponibilizada de forma livre pelo comitê olímpico, foi utilizada como base para os eventos esportivos, contendo data, local e descrição do esporte. Para os locais turísticos foram utilizadas a *API* do *Google Maps* (developers.google.com/maps/), com geolocalização e nome dos pontos turísticos da cidade do Rio de Janeiro, em seguida os locais recuperados foram enriquecidos com descrição semântica através da *API Babel Net* (babelnet.org).

Para obtenção dos dados dos usuários foi utilizada a rede social *Facebook* devido a sua popularidade e a *API* simplificada para a coleta de informações de perfil. A captura de contexto foi feita através da localização geográfica do usuário durante a recomendação, utilizando o GPS para dispositivos móveis e o IP para computadores.

Após a coleta nas bases externas os dados foram mantidos na base de dados do Mknob e acessados pelos módulos de recomendação e tradução. Para cada ação do usuário foi gerada uma pontuação usada para a gamificação do aplicativo. A Figura 14 mostra a arquitetura do aplicativo, evidenciando o Mknob com os módulos de recomendação e tradução que utilizam as informações do banco de dados, alimentado por dados dos usuários provenientes do *Facebook*, através do módulo de enriquecimento de perfil e enriquecimento semântico a partir de recursos externos com os locais que serão recomendados. Após o cálculo de recomendação os dados são enviados para o módulo de gamificação e em seguida apresentados ao usuário.

5.3.1.1 Modelo de Recomendação

O modelo de recomendação foi construído com base na arquitetura MMRecommender que propõe uma arquitetura aberta para a construção ou adaptação de sistemas de recomendação em diversos domínios de aplicação. A Figura 15 demonstra os componentes e as etapas do

método de recomendação implementadas no Mknob.

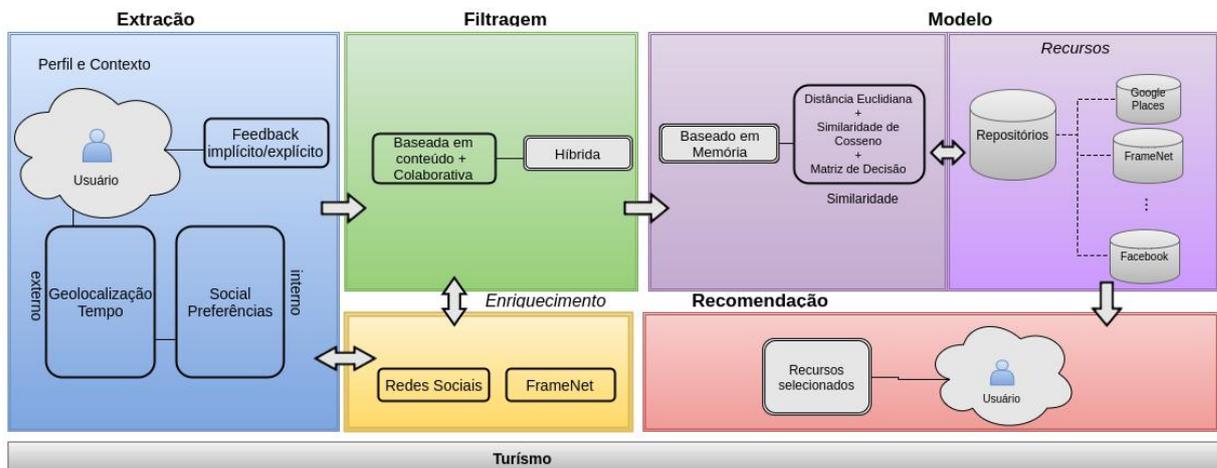


Figura 15 - Arquitetura MMRRecommender do aplicativo Mknob.

A arquitetura de recomendação Mknob é constituída por quatro etapas, sendo elas:

- **Extração:** Onde os dados de perfil e contexto do usuário são extraídos do *Facebook* e a geolocalização capturada através do GPS ou IP.
- **Filtragem:** Etapa onde os objetos recomendados e dados do perfil e contexto do usuário são agrupados. No Mknob foi utilizada uma filtragem híbrida, unindo a filtragem colaborativa e a baseada em conteúdo.
- **Enriquecimento:** É uma sub-etapa responsável pelo enriquecimento de características do perfil e contexto do usuário ou dos objetos recomendados. Essa sub-etapa foi implementada através dos *web-services* e *APIs* externas utilizados no Mknob.
- **Modelo:** É onde o cálculo de recomendação é realizado. No modelo do Mknob foi utilizado um modelo baseado em memória, ou seja, que utiliza toda a base de dados para efetuar a recomendação. O cálculo é realizado com base na similaridade de cosseno entre os objetos e o perfil do usuário. Caso exista informações de geolocalização também é acrescentado ao cálculo a distância euclidiana entre o usuário e os objetos. Por fim os dados são normalizados, ponderados e inseridos em uma matriz de decisão.
- **Recomendação:** É a etapa final do processo de recomendação, onde os objetos aderentes ao perfil do usuário são apresentados. No aplicativo é exibida uma lista ordenada, do mais aderente ao menos aderente, com 25 objetos.

5.3.1.2 *Processo de Recomendação*

O modelo de recomendação implementado pode variar o cálculo de recomendação de acordo com a informação do contexto do usuário, sendo dois cálculos possíveis: o primeiro considera apenas as preferências inferidas do usuário (perfil), e o segundo considera as preferências e a posição geográfica (contexto). A Figura 16 mostra o fluxo principal do aplicativo durante o processo de recomendação, também evidencia as *APIs* e *web-services* utilizados na extração de dados e enriquecimento semântico dos objetos recomendados. A Figura 17 exibe o processo completo de recomendação do aplicativo.

Montar vetor de características: É a etapa inicial do processo de recomendação, ativada quando o usuário faz uma busca textual ou por locais próximos, nesse momento o aplicativo consulta na base de dados o perfil e contexto atual do usuário e os objetos disponíveis. Esse método é conhecido como baseado em modelo e oferece recomendações atualizadas já que considera os dados mais atualizados para o processo de recomendação.

Após a consulta na base de dados é criado um vetor de características para o usuário e objetos. O vetor de características é uma abstração de dados que considera todas as características presentes nos objetos como colunas, e valores booleanos para a representação do estado daquela característica para um determinado usuário ou objeto. A Tabela 8 ilustra um exemplo de vetor de características.

Filtro Inclusivo: Em seguida é realizado um filtro inclusivo nos vetores dos objetos com base no vetor que representa o usuário. Esse filtro elimina apenas os objetos que possuem o valor '0' em todas as características com valor '1' do usuário. Com esse filtro é possível eliminar objetos com similaridade igual a '0', resultante em menos cálculos e conseqüentemente, uma melhora no desempenho. Em sequência é aplicado um algoritmo de comparação textual que mede a similaridade entre o texto pesquisado e os nomes dos objetos da base de dados. A Tabela 9 ilustra o vetor de característica após o filtro inclusivo.

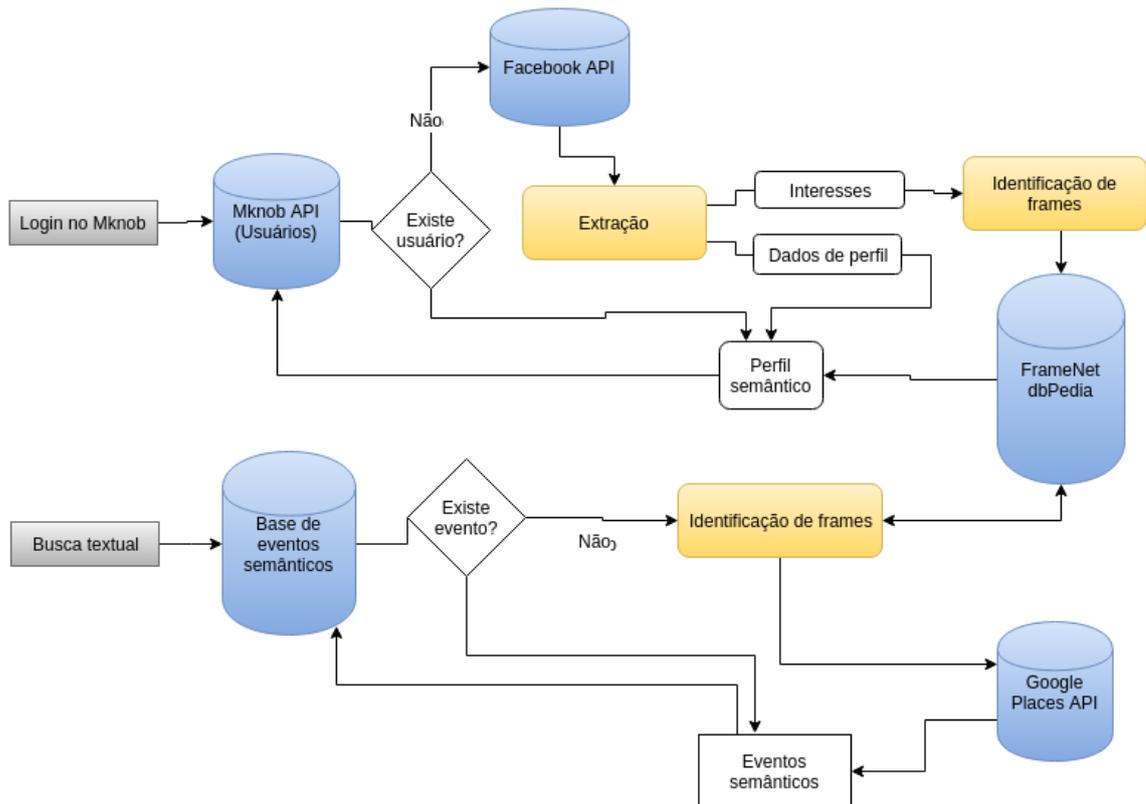


Figura 16 – Fluxo do aplicativo evidenciando as etapas e APIs usadas no processo de recomendação.

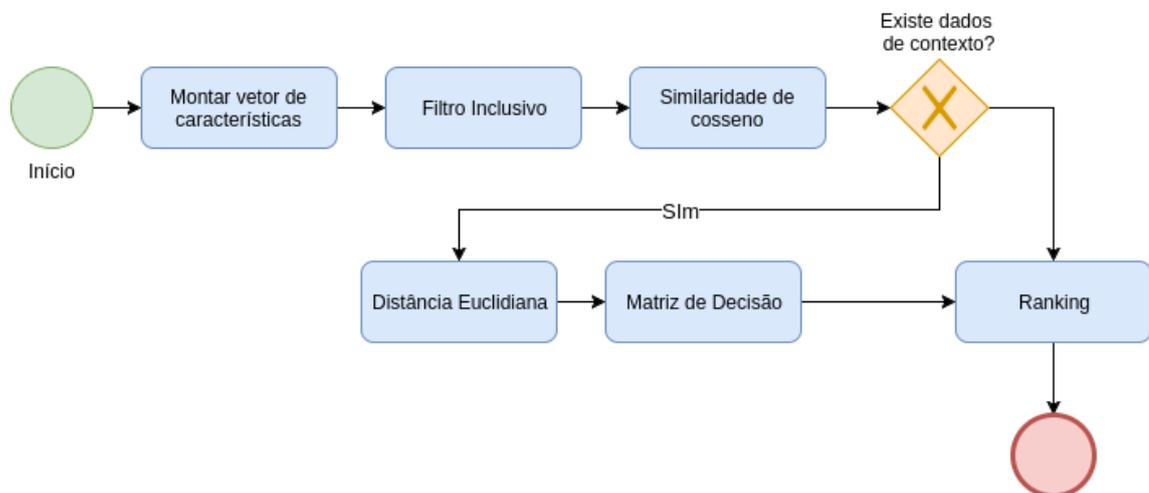


Figura 17 – Etapas do processo de recomendação usando na notação BPMN.

Similaridade de Cosseno: Etapa onde é calculada a similaridade de cosseno entre o usuário e os objetos restantes. A similaridade de cosseno mede o valor do cosseno do ângulo formado por dois vetores não nulos, podendo assumir valores reais entre -1 e 1. Essa medida é comumente utilizada como medida de similaridade em SRs, sendo que uma medida de similaridade igual a -1 significa que não existem características em comum, e similaridade igual a 1 implica em vetores, ou objetos, totalmente iguais.

	Compras	Fast-food	Café	Rugby
usuário	0	1	1	1
objeto 01	1	0	0	0
objeto 02	0	1	1	1
objeto 03	0	1	0	0

Tabela 8 - Exemplo de vetor de característica representando quatro características: *Compras*, *Fast-food*, *Café* e *Rugby*.

	Compras	Fast-food	Café	Rugby
usuário	0	1	1	1
objeto 01	1	0	0	0
objeto 02	0	1	1	1
objeto 03	0	1	0	0

Tabela 9 - Aplicação do filtro inclusivo e exclusão do objeto 01.

Distância Euclidiana: Esta etapa é iniciada apenas se existir dados de contexto (posição geográfica) do usuário. Inicialmente o vetor de característica já construído é acrescido com os valores de latitude e longitude. A Tabela 10 mostra um exemplo do novo vetor com dados geográficos.

Em seguida é calculada a distância euclidiana entre os usuários e todos os objetos. A distância euclidiana define entre pontos no plano cartesiano, nesse contexto os valores booleanos assumem valores dos eixos, sendo cada característica um eixo no plano cartesiano.

	Compras	Fast-food	Café	Rugby	Latitude	Longitude
usuário	0	1	1	1	-21.7662021	-43.3825507
objeto 02	0	1	1	1	-21.7662001	-43.3825514
objeto 03	0	1	0	0	-21.7662025	-43.3825501

Tabela 10 - Vetor de característica formado com os dados geográficos.*

*O objeto 01 não está presente no vetor porque foi removido na etapa anterior (filtro inclusivo).

Matriz de Decisão: Após o cálculo da distância euclidiana existem duas medidas para o cálculo de recomendação, sendo a similaridade de cosseno que define o nível de similaridade entre o usuário e os objetos, e a distância euclidiana, que de modo simplificado, define a distância geográfica do usuário aos objetos.

Como as duas medidas possuem comportamentos distintos é necessário realizar uma normalização dos dados para uma possível comparação, nesse cenário os valores da similaridade de cosseno e distância euclidiana de cada item foram divididos pelo maior valor obtido de cada métrica e ponderado por valores obtidos a partir de testes empíricos com o

dataset utilizado na etapa 1 do estudo de caso 3, que será explicado posteriormente, desta forma foram definidos os valores de 0.7 para a similaridade de cosseno e 0.3 para a distância euclidiana. Nos testes empíricos estes valores mantiveram os objetos mais similares ao usuário no topo da lista e utilizando a distância como critério de desempate. Após esse processo os valores foram inseridos em uma matriz de decisão.

Ranking: Ao fim do processo os objetos são ordenados de forma decrescente pelos valores obtidos nas etapas anteriores.

5.3.1.3 Comparação Textual

Existem diversas técnicas de comparação textual a nível sintático, muitas delas utilizam abordagens diferentes e para o mesmo cenário podem apresentar valores distintos, por isso é necessário entender o cenário de comparação, ou seja, os textos que serão comparados e o valor esperado.

Durante o desenvolvimento do aplicativo foi realizada uma pesquisa *ad-hoc* para reunir as principais técnicas de comparação textual, sendo elas: *Levenshtein*, *Needleman Wunch*, *Smith-Waterman*, *Smith-Waterman Gotoh*, *Smith-Waterman Gotoh Windowed Affine*, *Jaro*, *Jaro Winkler*, *n-Grams Distance*, *Block Distance*, *Cosine Similarity*, *Euclidean Distance*, *Chapman Length Deviation*, *Overlap Coefficient*.

Em seguida essas técnicas foram analisadas para identificar qual é a mais aderente a possíveis cenários de busca no aplicativo, além de menor esforço computacional. Com base nos resultados a técnica *n-Grams* foi escolhida para incrementar a etapa do filtro inclusivo.

O algoritmo *n-Grams* divide os textos que serão comparados em unidades com ‘n’ caracteres, chamados de ‘grams’, em seguida é comparada a quantidade de ‘grams’ similares nos textos comparados, quanto maior a quantidade de ‘grams’ maior a similaridade. Para o aplicativo foi escolhido o valor de n igual a três, ou seja, os textos foram divididos em ‘grams’ com três caracteres.

A Figura 18 mostra os métodos implementados em PHP que medem a similaridade entre dois textos através do algoritmo *n-grams*. O primeiro método ‘*nGrams*’ faz a contagem de ‘grams’ semelhantes entres os textos, e o método ‘*Trigrams*’ separa o texto passado como parâmetro em ‘grams’ de três caracteres.

A Figura 19 mostra a implementação do algoritmo que calcula a similaridade de cosseno entre o vetor de características do usuário e os objetos. O algoritmo realiza o cálculo

do cosseno entre o ângulo formado pelos pares [usuário, objeto 1], [usuário, objeto 2] e assim por diante, conforme mostrado na equação 4 .

Para o cálculo da distância euclidiana são usados dados da localização geográfica do usuário e dos objetos da base de dados, utilizando longitude e latitude. O processo para medir a distância é feito pelos pares [usuário, objeto 1], [usuário, objeto 2] e assim por diante. Dessa forma a distância euclidiana entre os pontos é medida utilizando um plano cartesiano com duas coordenadas. A Figura 20 mostra a implementação em PHP conforme a equação 5 de distância euclidiana.

5.3.2 Desenvolvimento do Sistema

A equipe de desenvolvimento do aplicativo contou com quatro desenvolvedores entre *front-end*, *back-end* e *fullstacks*. Para o desenvolvimento *front-end* foi utilizado o *framework Materialize CSS*, *HTML 5*, *CSS3*, *Javascript* e *AngularJS*. No *back-end* foi utilizada a linguagem PHP com *framework Maestro*, desenvolvido pela UFJF que implementa o padrão MVC (*Model-View-Controller*), fortemente baseado em serviços *RESTful*. Também foram utilizadas *APIs* que auxiliaram processos internos do aplicativo, sendo elas:

- *Google Places API*, responsável pela captura dos dados de localização de pontos turísticos;
- *Olympic Data Feed API*, com informações detalhadas dos jogos olímpicos, com data, hora e local;
- *API FrameNet* que fez o enriquecimento semântico de locais turísticos com base na abordagem de Frames;
- *Facebook Login API*, responsável pela captura dos dados de perfil do usuário com base nas informações do Facebook.

Para fazer a persistência de dados foi adotado o banco de dados *mySQL* hospedado em um servidor da Universidade Federal de Juiz de Fora. A Figura 21 mostra o modelo entidade-relacional do banco de dados.

```

static function nGrams($str1, $str2){
    $textSearch = array();
    $textSearch = MathEngineService::Trigrams($str1);
    $venueName = array();
    $venueName = MathEngineService::Trigrams($str2);

    $score = 0;

    for ($i = 0 ; $i < count($textSearch) ; $i ++) {
        for ($j = 0 ; $j < count($venueName) ; $j ++) {
            if ($textSearch[$i] == $venueName[$j])
                $score ++;
        }
    }
    return ($score/(count($textSearch) + count($venueName)));
}

static function Trigrams($word){
    $ngrams = array();
    $len = strlen($word);
    for ($i = 0 ; $i + 2 < $len ; $i ++) {
        $ngrams[$i] = $word[$i] . $word[$i + 1] . $word[$i + 2];
    }

    return $ngrams;
}

```

Figura 18 – Algoritmo em PHP dos métodos nGrams e Trigrams.

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \quad (4)$$

$$d(user, obj) = \sqrt{(user_x - obj_x)^2 + (user_y - obj_y)^2} \quad (5)$$

```

static function cosine($itens, $user)
{
    $itensSimilarity = MathEngineService::cosineSlice($itens);
    $userSimilarity = MathEngineService::userCosineSlice($user);
    $dot_Product = MathEngineService::dotProduct($userSimilarity, $itensSimilarity);
    $dot_Product2 = sqrt(MathEngineService::dotProduct($userSimilarity, $userSimilarity)) *
    sqrt(MathEngineService::dotProduct($itensSimilarity, $itensSimilarity));
    $cosine = round($dot_Product / $dot_Product2, 2);

    return $cosine;
}

```

Figura 19 – Algoritmo em PHP que calcula a similaridade de cosseno.

```

static function euclidean($itens, $user)
{
    $itensSimilarity = MathEngineService::euclideanSlice($itens);
    $userSimilarity = MathEngineService::userEuclideanSlice($user);
    $sum = pow($userSimilarity[0] - $itensSimilarity[0], 2) +
    pow($userSimilarity[1] - $itensSimilarity[1], 2);
    $euclidean = sqrt($sum);

    return $euclidean;
}

```

Figura 20 – Algoritmo em PHP que calcula a distância Euclidiana.

5.3.2.1 Integração com Facebook

A rede social *Facebook* foi escolhida para efetuar o controle de *login* e captura dos dados do usuário. Nesse contexto a rede social oferece uma SDK (*Software Development Kit*) simples e robusta para capturar informações utilizadas no *login*, como *e-mail* e nome, e dados de inferência de interesses como páginas curtidas.

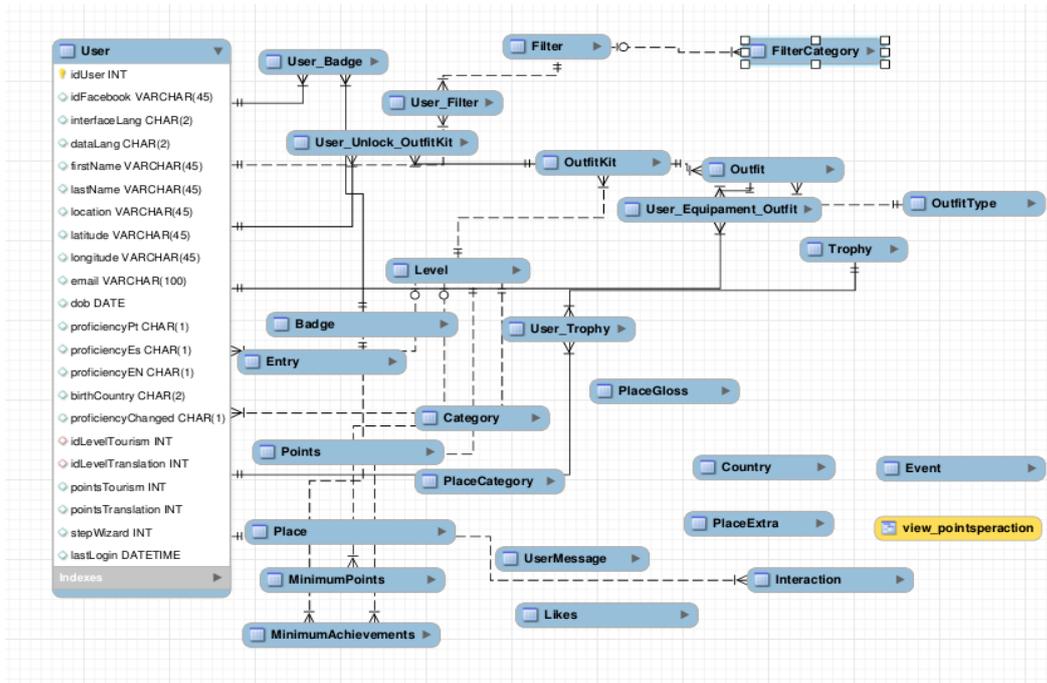


Figura 21 – Modelo entidade relacionamento do banco de dados utilizado no Mknob.

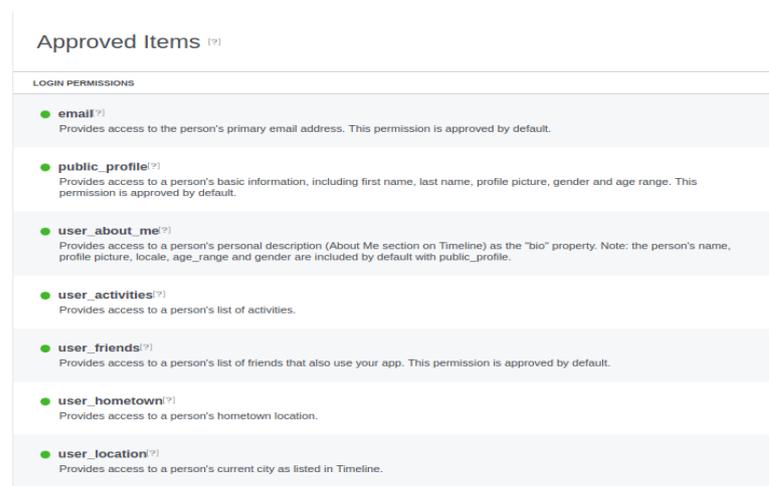


Figura 22 – Lista de permissões aprovadas pelo *Facebook*.

Através da SDK do *Facebook* (versão 2.5) importada para o aplicativo Mknob foi possível acessar informações do usuário utilizando código *JavaScript* e em seguida enviando para um *web-service* do aplicativo que faz a persistência dos dados no banco de dados

mySQL. Após a importação da SDK para o projeto Mknob foi criada uma conta de desenvolvedor na rede social, e em seguida um aplicativo do *Facebook* referente ao Mknob.

Para a publicação do aplicativo no *Facebook* foi necessário cumprir os requisitos, detalhando quais permissões do usuário o aplicativo irá utilizar, como essas informações serão usadas e enviar um vídeo do Mknob utilizando os dados. Para o projeto foram solicitados os dados de: Nome, *e-mail*, descrição do perfil, foto do perfil, lista de amigos, atividades do usuário, cidade natal e localização (Figura 22). Ao fim do processo o aplicativo foi publicado e disponibilizado aos usuários da rede social.

Através da SDK do *Facebook* também foi possível adicionar a opção de compartilhar, onde o usuário pode compartilhar o objeto recomendado diretamente no *feed* de notícias da rede social.

5.3.2.2 Coleta de Dados

A coleta de dados foi feita a partir de duas fontes distintas sendo à base de dados do aplicativo Mknob e dados gerados pelo *API* do *Facebook*. A base de dados do aplicativo foi implementada em um banco relacional e armazenadas informações de cadastro do usuário como nome, idade e *e-mail*, além de informação de usabilidade, como pesquisas realizadas. A base do *Facebook* exibe informações demográficas dos usuários a partir dos dados preenchidos no perfil do *Facebook* e previamente liberados para a consulta durante a etapa de cadastro, além de ações realizadas no *Facebook* relacionadas ao aplicativo. Desta forma é possível dividir os tipos de dados em dois grupos, sendo eles:

- Dados implícitos (*Feedback Implícito*): Buscas por locais e eventos esportivos,
- Dados explícitos (*Feedback Explícito*): Avaliação positiva na forma de ‘like’ utilizada em redes sociais e o compartilhamento através do *Facebook* de locais ou eventos esportivos.

Em um sistema de recomendação é importante definir o perfil e contexto do usuário que irá receber a recomendação. Esse perfil e contexto foi traçado durante a fase de projeto do aplicativo e norteou a construção do algoritmo de recomendação durante a fase de implementação.

- Contexto: Usuários localizados na cidade do Rio de Janeiro entre os dias 05 e 21 de Agosto, com acesso a *Internet* móvel através de dispositivos móveis.
- Perfil: Turistas interessados nos jogos olímpicos e em locais turísticos

aderentes ao seu perfil.

Durante o evento o aplicativo estava disponível publicamente através do endereço na *Internet*, portanto não houve pré-seleção da amostra, e todos os participantes utilizaram o aplicativo de forma voluntária.

5.3.2.3 Fonte de Dados

A *API* do *Facebook* oferece ferramentas para captura dos dados de interação (*feedback* implícito) e quantidades de usuários ativos por dia, semana ou mês, também é possível utilizar ferramentas de dados demográficos avançadas do *Facebook*. As ferramentas de dados demográficos não foram utilizadas devido a regras mais complexas de publicação na rede social. Através do ‘*dashboard*’ do aplicativo no *Facebook* os dados foram coletados e agrupados para a análise.

5.3.3 Estudo de Caso III

O estudo de Caso III tem escopo definido como: analisar o sistema de recomendação baseado na arquitetura *MMRecommender*, com o propósito de avaliar, com relação à eficácia, sob o ponto de vista do usuário, as recomendações no contexto dos jogos olímpicos Rio 2016.

Considerando que o SR para o Projeto *Mknob* consiste de um projeto em tempo real, com usuários diversificados e aleatórios, a avaliação foi feita em duas etapas.

ETAPA 1

A primeira etapa para medir a eficácia do modelo de recomendação foi realizada através de um experimento controlado em laboratório. No experimento foi utilizado o *dataset* com 277 locais turísticos enriquecidos semanticamente a partir da *FrameNet* com coordenadas geográficas obtidas através da *API* do *Google Maps*.

Para realizar a recomendação é necessário um usuário alvo, ou seja, o usuário irá receber os itens recomendados, por isso, nessa primeira etapa do experimento, foi criado um perfil de usuário fictício para receber a recomendação e também foi simulada a sua posição geográfica. Em seguida os dados do usuário foram enviados para um *web-service* implementado em PHP que faz a leitura do banco de dados do experimento (*dataset*) e realiza os cálculos de recomendação com base no perfil e localização do usuário.

Para medir os resultados foram usadas as métricas *Precision*, *Recall*, *F1-Score* e *G-*

Measure, onde TP são valores verdadeiros positivos (*True Positives*), FP falso positivo (*False Positive*) e FN falso negativo (*False Negative*). Tais métricas foram encontradas a partir do mapeamento sistemático apresentado anteriormente neste trabalho.

- *Precision*: É a razão de objetos recuperados que são relevantes, e é representada pela equação 6.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

- *Recall*: É a fração de documentos que são relevantes para a consulta e foram recuperados, conforme apresentado na equação 7.

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

- *F1-Score*: Também conhecida como *F-measure* é a média harmônica entre os valores de *precision* e *recall*. A equação 8 demonstra a *F1-Score*.

$$F1\ score = \frac{2*(Precision*Recall)}{Precision+Recall} \quad (8)$$

- *G-Measure*: É a média geométrica entre os valores de *precision* e *recall* conforme mostra a equação 9.

$$gmeasure = \sqrt{Precision * Recall} \quad (9)$$

Para utilizar as métricas de *Precision* e *Recall* é necessário conhecer quais objetos devem ser recuperados, e no contexto deste trabalho significa identificar os locais que o usuário deve receber como recomendação. Esta análise foi feita com base no perfil do usuário criado para o experimento e posteriormente listados em uma planilha eletrônica.

Para este experimento foram escolhidos os 59 locais mais aderentes ao perfil do usuário e após o processamento do modelo de recomendação os 59 itens iniciais da lista foram comparados com os locais selecionados.

O tempo de execução do algoritmo e consumo de memória não foram medidos por serem métricas que não podem avaliar a eficácia do modelo de recomendação e sim a eficiência, e medir a eficiência do modelo está fora do escopo dessa pesquisa.

Com o ambiente do experimento preparado, isto é, *dataset* e perfil do usuário definidos, *web-service* implementado e métricas de avaliação definidas o experimento foi

executado e os dados coletados. Os resultados do experimento são apresentados na Tabela 11.

Métrica de Avaliação	Resultado
<i>Precision</i>	96.552%
<i>Recall</i>	98.276%
<i>F1 Score</i>	97.406%
<i>G measure</i>	97.410%

Tabela 11 - Resultado do modelo de recomendação.

Evidências observadas na Etapa 1

O experimento possui limitações quanto ao tamanho do *dataset* analisado e suas variações. Devido ao curto espaço de tempo para desenvolvimento do aplicativo e o esforço gerado durante a fase de enriquecimento dos dados foi disponibilizado um *dataset* inicial com 277 locais da cidade de Juiz de Fora – Minas Gerais. Com base nesse único *dataset* o experimento foi executado e avaliado.

Outra possível limitação está relacionada ao viés durante a criação do perfil usado no experimento, que foi criado com base nos interesses de uma pessoa escolhida de forma aleatória. Essa escolha é justificada pela característica do aplicativo que não possui um perfil de usuário bem definido.

ETAPA 2

A segunda etapa consistiu de um estudo de caso com o sistema de recomendação liberado para usuários externos, não controlados e aleatórios. Nesse contexto foram coletados dados para a avaliação do sistema desenvolvido para o projeto Mknob.

O

Gráfico 10 mostra a quantidade de usuários ativos por dia, definida pela quantidade de usuários diferentes que fizeram o *login* no aplicativo naquele dia.

O Gráfico 11 exhibe o total de dispositivos de acesso por dia, durante 9 dias, dividido em dois grupos sendo:

- o *Mobile Web*: Dispositivos móveis como *smartphones* e *tablets*,
- o *Web*: Através de um navegador de *internet*, ou seja, utilizando computadores e *notebooks*.

Usuários Ativos

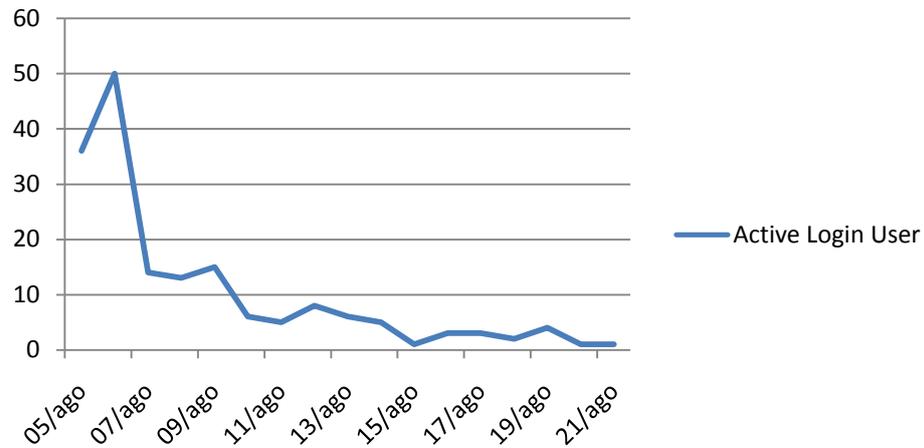


Gráfico 10 - Quantidade de usuários ativos por dia.

Dispositivo de Acesso

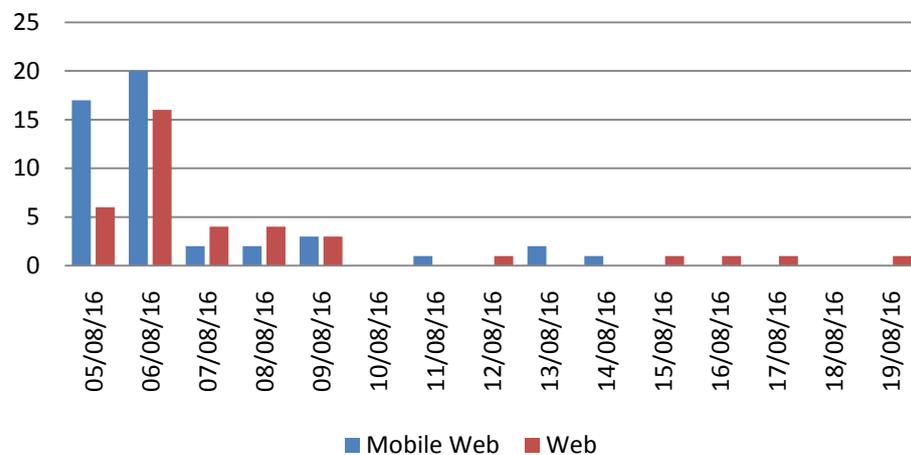


Gráfico 11 - Total de dispositivos de acesso por dia.

Através dos gráficos apresentados é possível perceber uma queda na quantidade de acessos ao longo do tempo, que pode ser justificada pela falta de divulgação e campanhas de *marketing* nas redes sociais. O aplicativo foi divulgado para o meio acadêmico, na página da Universidade Federal de Juiz de Fora e em uma emissora local um dia antes do lançamento (dia 04/08), porém os veículos de comunicação usados não estão aderentes ao perfil e contexto do usuário do aplicativo. Dessa forma podemos inferir que a alta quantidade de usuários nos dois dias foram pessoas impactadas pelos meios de comunicação onde o

aplicativo foi divulgado.

Após a extração de dados do *Facebook* foi realizado um agrupamento dos dados extraídos e com a base de dados do aplicativo que capturou a avaliação explícita do usuário quanto ao local recomendado. Essa avaliação foi feita na forma de ‘like’, utilizando o mesmo princípio do *Facebook*.

Com base nos dados coletados foi criado o indicador ‘*positive actions*’ que é a soma do total de ‘likes’ e ‘shares’ (compartilhamento). Com base nesse indicador espera-se inferir ações positivas do usuário quanto ao local recomendado. Essas ações positivas foram consideradas como avaliações positivas já que, se o usuário ‘curtiu’ ou compartilhou um local, significa que, possivelmente, aquele local está aderente ao perfil e contexto desse usuário.

A Tabela 12 mostra a quantidade total de usuários diferentes ativos por dia e a quantidade de ‘likes’, ‘shares’ e ‘*positive actions*’ por dia. Através dela é possível perceber que a quantidade de usuários por dia não implica numa maior interação com o aplicativo.

Tabela 12 - Quantidade de usuários ativos e ações positivas por dia.

Dia	<i>Active Login User</i>	<i>Likes</i>	<i>Shares</i>	<i>Positive Actions</i>
5-Aug	36	27	3	30
6-Aug	50	7	0	7
7-Aug	14	3	0	3
8-Aug	13	2	0	2
9-Aug	15	4	19	23
10-Aug	6	0	0	0
11-Aug	5	0	0	0
12-Aug	8	1	0	1
13-Aug	6	2	1	3
14-Aug	5	0	1	1
15-Aug	1	0	0	0
16-Aug	3	0	0	0
17-Aug	3	0	0	0
18-Aug	2	0	0	0
19-Aug	4	1	1	2
20-Aug	1	0	0	0
21-Aug	1	0	0	0

O Gráfico 12 é a representação visual da Tabela 12 e através dele é possível notar que no dia 9 de Agosto o total de interações no aplicativo ultrapassou o total de usuários ativos.

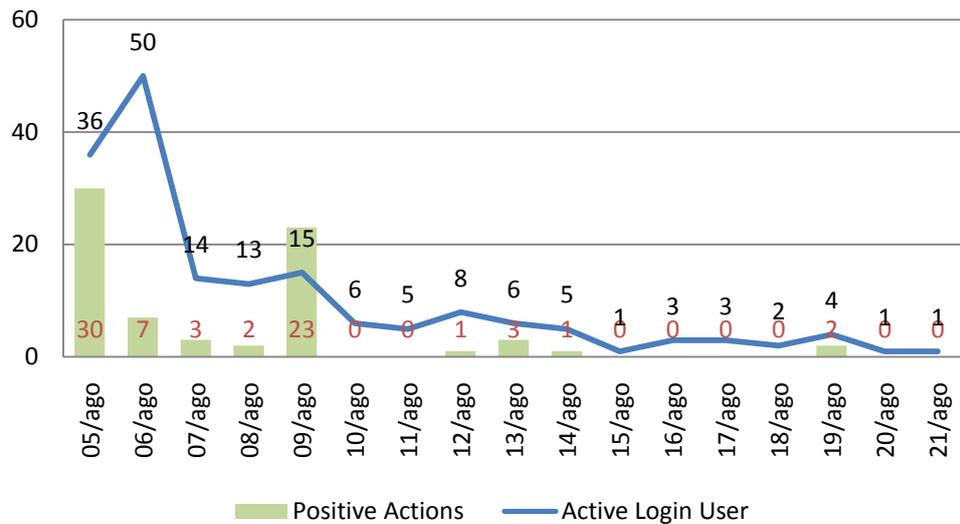


Gráfico 12 - Representação gráfica do total de usuários ativos e ações positivas por dia.

O Gráfico 13 exibe a quantidade de pesquisas textuais realizadas por dia dentro da aplicação. Através dela é possível perceber que o dia de lançamento (5 de Agosto) e o dia 9 de Agosto foram os dias com usuários com maior interação.

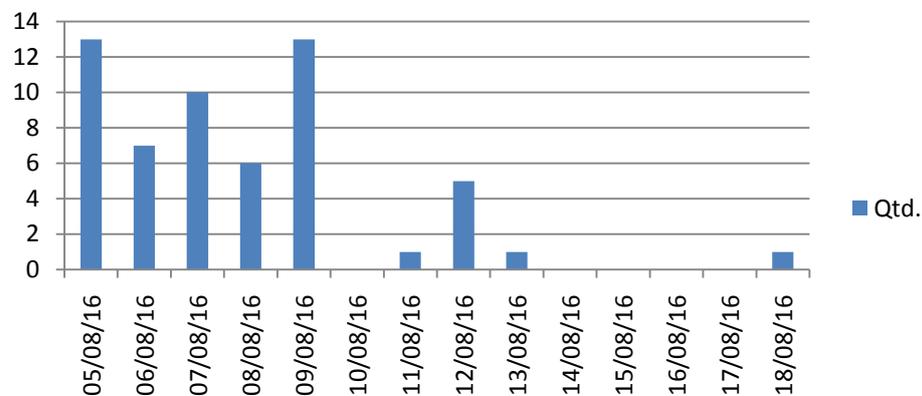


Gráfico 13 - Quantidade de pesquisas textuais realizadas no aplicativo por dia.

O comportamento do usuário com base nas pesquisas textuais e ‘likes’ foram avaliados por turnos (manhã, tarde e noite) e por dia, com a finalidade de entender o tipo de comportamento do usuário com base no período mais ativo. O Gráfico 14 e Gráfico 15 mostram o comportamento do usuário por período e dia com base nas pesquisas textuais e ‘likes’ respectivamente.

Através dos dados capturados pelo *Facebook* é possível identificar um perfil de usuário do aplicativo. O Gráfico 16 exibe os termos mais buscados, que apesar de existirem termos diretos como ‘tênis’, ‘cerveja’, entre outros, também foram identificados termos que

remetem a uma ação, como ‘quero comer pizza’. A presença desses termos pode sugerir que os usuários preferem buscar por termos que remetem a uma ação explícita.

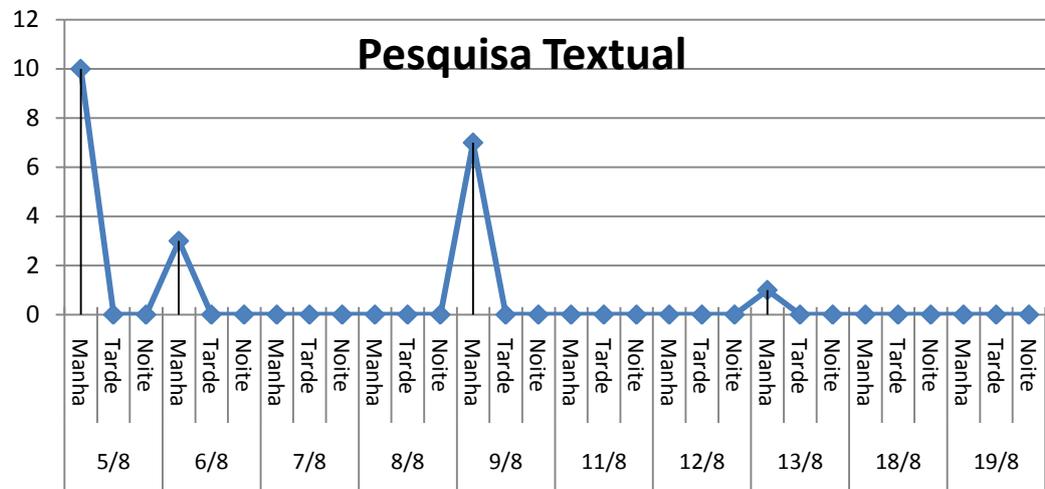


Gráfico 14 - Perfil de comportamento com base nas pesquisas textuais.

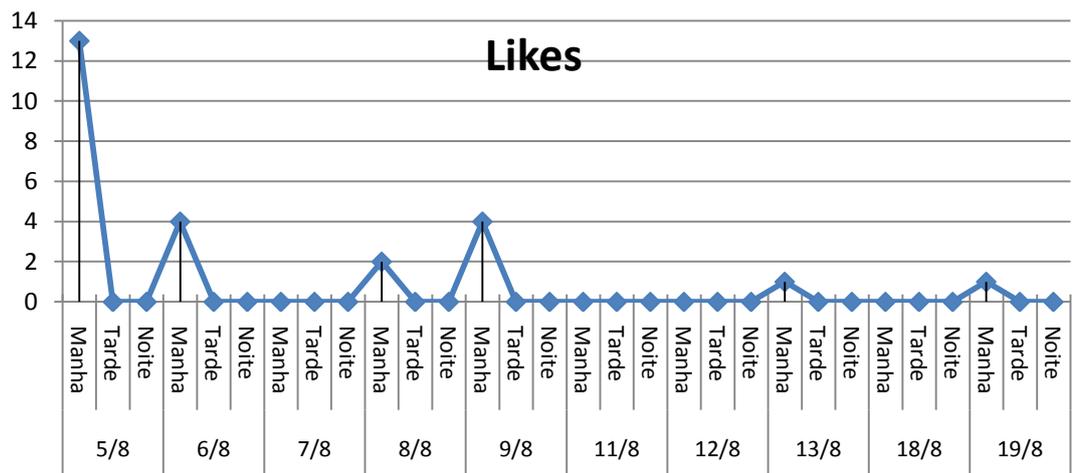


Gráfico 15 - Perfil de comportamento com base nos 'likes'.

Buscas textuais

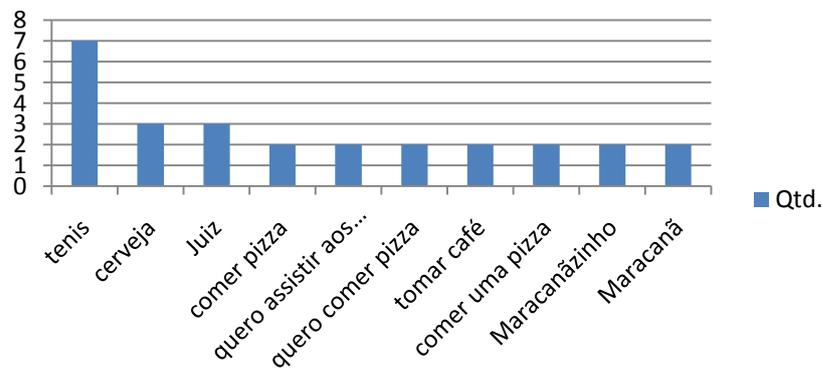


Gráfico 16 - Buscas mais frequentes.

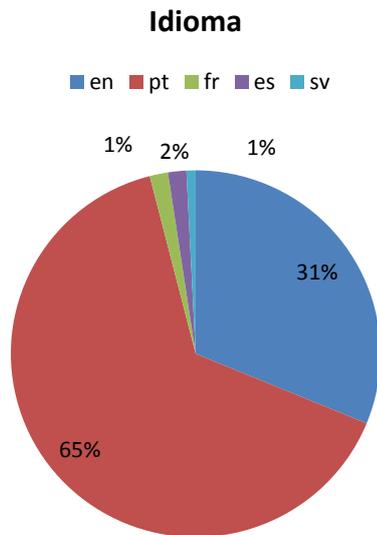


Gráfico 17- Percentual de idiomas.

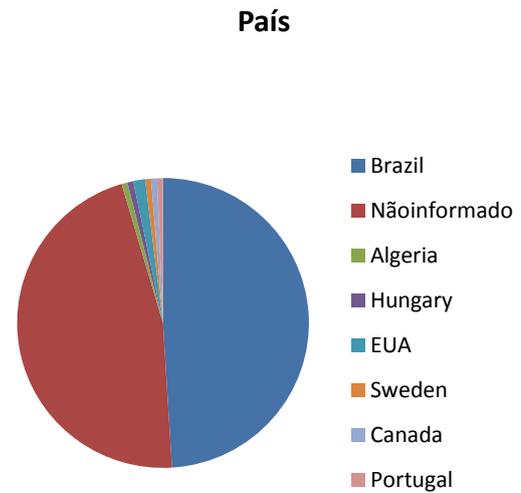


Gráfico 18- Percentual por país.

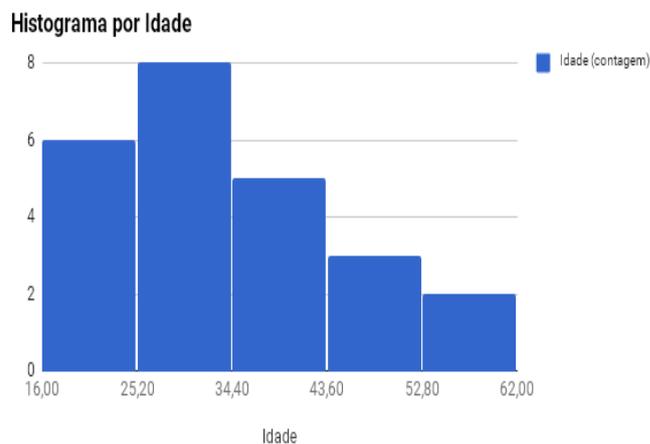


Gráfico 19 - Histograma de idade.

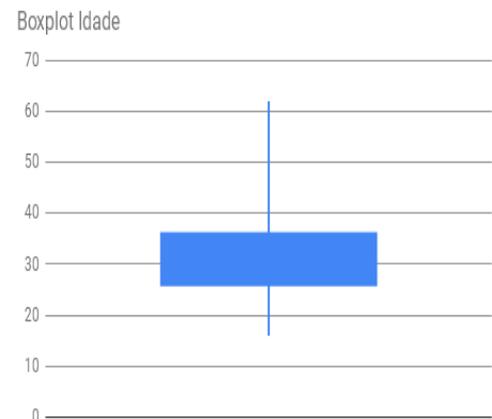


Gráfico 20 - Boxplot por idade.

Também é possível identificar o país de origem e idioma principal dos usuários do aplicativo, o Gráfico 16 e Gráfico 17 mostram que a maioria dos participantes são brasileiros, com idioma principal português, o que é justificado pelo local do evento. O Gráfico 18 e Tabela 13 exibem a relação de país com o idioma principal. Através dela é possível notar que a maior diversidade de idiomas está localizada no Brasil, sendo o português, inglês, espanhol e francês. Com base nesta informação pode-se inferir que alguns usuários utilizam o *Facebook* em outros idiomas diferentes do português.

Os Gráfico 19 e Gráfico 20 mostram o histograma e o *boxplot* por idade respectivamente. A idade foi calculada com base no ano de nascimento informado no *Facebook*, desconsiderando o dia e mês.

Tabela 13 - Relação país de origem e idioma principal.

País	Idioma	Qtd.
Brazil	en	20
	pt	52
	fr	1
	es	2
Algeria	pt	1
Canada	fr	1
EUA	pt	1
EUA	en	1
Hungary	en	1
Portugal	en	1
Sweden	sv	1
Não informado	en	38
	pt	33

Evidências Observadas na Etapa 2

O estudo de caso III possui limitações que podem influenciar os resultados do mesmo, como por exemplo a veracidade dos dados informados pelos usuários ou a liberação das informações, ou seja, os dados extraídos do *Facebook* podem não corresponderem à realidade do usuário, como país de origem e idioma principal. Outro detalhe está relacionado à possível existência de perfis falsos, que a princípio não foram identificados, porém é uma característica da rede social e tendo em vista que o aplicativo é liberado para qualquer usuário é possível existir perfis falsos na base de dados analisada.

Também existem limitações quanto à veracidade dos dados explícitos informados pelos usuários que podem ou não corresponder a realidade. Após o primeiro *login* no aplicativo o usuário é direcionado para uma tela onde é possível completar seu cadastro informando interesses pessoais, porém esta tela é opcional e possui muitos itens, o que pode desmotivar o preenchimento do cadastro.

5.3.4 Considerações Finais da Seção

Esta seção descreveu a avaliação do aplicativo Mknob, que pode ser dividida em duas etapas, a primeira onde foram feitas avaliações no modelo de recomendação, construído com base na arquitetura aberta MMRecommender. Nesta primeira etapa foram utilizados métodos estatísticos como *Precision*, *Recall*, *F1-Score* e *G-Measure*, comuns em trabalhos de SR, e identificados no mapeamento sistemático apresentado no Capítulo 3.

Na segunda etapa foi realizada uma coleta de dados reais dos usuários do aplicativo com base na utilização do sistema e dados de *login* do *Facebook*. Estes dados foram analisados a fim de identificar padrões de comportamento e perfil de usuários do aplicativo.

Nas duas etapas é possível entender que o aplicativo Mknob cumpriu seu objetivo em recomendar pontos turísticos e eventos esportivos durante a olimpíada do RIO 2016.

Outro ponto observado durante o processo de implementação do estudo de caso aplicado ao turismo foi que o modelo de arquitetura gerado contribuiu para a comunicação entre diferentes grupos envolvidos no projeto.

6 CONSIDERAÇÕES FINAIS

Existem múltiplas aplicações de sistemas de recomendação, o que dificulta a existência de classificações e taxonomias na área. Sob essa ótica foi proposta uma arquitetura denominada MMRecomender, para apoiar o desenvolvimento de SRs em diversos domínios.

Um projeto de pesquisa científica possui sete características fundamentais descritas por (DRESCH; LACERDA; ANTUNES, 2015apud HAVNER *et al.*, 2014), que consistem na (I) criação de um artefato para (II) atender a um problema em particular, (III) cuja utilidade deve ser explicitada através de uma avaliação apropriada de sua aplicabilidade e (IV) os resultados e contribuições da pesquisa devem ser compartilhado com os profissionais interessados e a academia. Para assegurar sua validade, (V) as investigações devem ser conduzidas com rigor e (VI) as possíveis formas de solução analisadas e, por fim, (VII) os resultados devem ser comunicados aos interessados.

Dessa forma, esta pesquisa atende aos sete critérios fundamentais ao propor (I) a arquitetura aberta MMRecommender para (II) favorecer o planejamento, desenvolvimento, compartilhamento e reuso de modelos de Sistemas de Recomendação, (III) avaliá-la por meio de estudos de caso em diferentes domínios e com objetivos diferenciados e demonstrar a viabilidade técnica da proposta e dos conceitos e componentes envolvidos em contextos reais de utilização, e (IV) apresentar os resultados à comunidade acadêmica e aos profissionais interessados através da presente dissertação, dos sistemas disponibilizados na Web e das publicações. Foi mantido (V) o rigor metodológico durante o desenvolvimento da pesquisa, (VI) analisando as soluções e alternativas existentes e (VII) garantindo a publicação dos resultados.

6.1 CONTRIBUIÇÕES

A contribuição principal está na arquitetura aberta MMRecommender que, como evidenciado neste trabalho, pode auxiliar no processo de construção de modelos de recomendação, podendo ser utilizados como módulos, serviços ou camadas aplicados a sistemas que utilizam a recomendação como meio ou como parte principal em um sistema que utiliza a recomendação como fim.

Em comparação com outros trabalhos a arquitetura apresentada propõe um modelo bem definido para o desenvolvimento de módulos de recomendação ou SRs, onde as etapas e

sub-etapas e seus componentes estão detalhados, contribuindo para as fases de identificação de requisitos, análise e projeto no processo de desenvolvimento.

Como contribuição deste trabalho pode-se citar, além da arquitetura proposta, o mapeamento sistemático que oferece uma visão do estado da arte em SR que pode ser utilizado por pesquisadores e interessados na área. Os estudos de casos também podem ser utilizados como uma fonte de referência de implementação e modelos de recomendação aplicados a diferentes domínios. Nesse ponto de vista o protocolo de avaliação dos estudos de casos também podem servir como base para a avaliação de outros SR.

6.2 LIMITAÇÕES

Além das limitações mencionados anteriormente referentes a cada estudo de caso a arquitetura MMRecommender também possui limitações, como exemplo podemos citar a sua base teórica proveniente de trabalhos identificados no mapeamento sistemático e experiência do grupo de pesquisa.

Outro ponto importante que a arquitetura proposta não abordou é a presença de uma etapa de avaliação do modelo de recomendação instanciado. Esta etapa pode incluir componentes e métricas identificados comuns para a avaliação de SR. Tais componentes foram identificados no mapeamento sistemático porém não faz parte do escopo deste trabalho analisar o processo de avaliação de modelos de recomendação, apesar de termos incluído essa etapa no estudo de caso III.

6.3 TRABALHOS FUTUROS

A partir da arquitetura aberta MMRecommender existem muitas possibilidades a serem exploradas, uma delas seria a inclusão de uma etapa de avaliação do modelo gerado com a arquitetura, essa etapa pode auxiliar a escolha da métrica de avaliação mais aderente com base nos itens escolhidos nas etapas anteriores do arquitetura.

Durante a extração de dados dos trabalhos incluídos no mapeamento sistemático e a identificação do estado da prática através de comunidades e grupos de pesquisas relacionados ao tema foi possível identificar a evolução de SR. Essa evolução pode ser categorizada como dupla-recomendação, que aborda a dinâmica em identificar pessoas com perfis e interesses comuns para a formação de pares ou grupos.

Sob perspectiva de implementação, a arquitetura proposta pode evoluir para a criação de um *framework* para facilitar o desenvolvimento de modelos de SR. Tal *framework* pode abordar todas as etapas da arquitetura, oferecendo, por exemplo, uma série de algoritmos implementados e a possibilidade de incrementar novos algoritmos como *web-services*. Outro ponto interessante é o desenvolvimento de uma camada de testes de *software* que podem executar métricas de avaliação do SR automaticamente, esta camada pode evoluir para uma *API* de avaliação podendo ser utilizada em outros softwares.

Em relação a SR é possível explorar suas aplicações em redes sociais aplicando recomendação na formação de grupos ou perfis em cada rede social e utilizando o grande volume de dados inseridos explorando ainda mais a utilização do feedback implícito e métodos de enriquecimento de dados. Também é possível explorar o desempenho de SR em relação ao tempo para produzir recomendações, este ponto é importante no contexto de sistemas que precisam entregar respostas em um curto espaço de tempo.

Por fim, o grupo de pesquisa NEnC pretende evoluir o processo de desenvolvimento da arquitetura proposta para definir uma plataforma de ecossistema de *software* de recomendação, explicitando os fornecedores e caminhos de evolução do mesmo.

REFERÊNCIAS BIBLIOGRÁFICAS

- ABDALLA, A. et al. R.ECOS - Educational Recommender Ecosystem. **Proceedings - 2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, JSOS 2017**, n. May, p. 48–54, 2017.
- ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. **IEEE Transactions on Knowledge and Data Engineering**, v. 17, n. 6, p. 734–749, 2005.
- ALMEIDA, L. A. et al. A Case Study on the Usage of the Value Blueprint for Ecosystem Design. p. 431–438, 2015.
- ALMEIDA, R. F.; CAMPOS, F.; STROELE, V. Sistemas de Recomendação de Recursos Educacionais para Grupos de Redes Sociais: um Mapeamento Sistemático. **Anais do Simpósio Brasileiro de Informática na Educação (SBIE)**, n. Sbie, p. 1022–1031, 2015.
- ALSPAUGH, T. A.; ASUNCION, H. U.; SCACCHI, W. The challenge of heterogeneously-licensed systems in open architecture software ecosystems. **Software ecosystems: Analyzing and managing business networks in the software industry**, p. 103–120, 2013.
- BADARO, G. et al. **A hybrid approach with collaborative filtering for recommender systems**. Wireless Communications and Mobile Computing Conference (IWCMC). **Anais...2013**
- BOBADILLA, J. et al. Recommender systems survey. **Knowledge-Based Systems**, v. 46, p. 109–132, 2013.
- BURKE, R. Hybrid recommender systems: Survey and experiments. **User Modeling and User-Adapted Interaction**, 2002.
- CARVALHO, L. A M. C.; MACEDO, H. T. Introdução aos sistemas de recomendação para grupos. **Revista de Informática Teórica e Aplicada**, v. 21, p. 78–109, 2014.
- CASAGRANDE, M. F. R.; KOZIMA, G.; WILLRICH, R. Técnica de Recomendação Baseada em Metadados para Repositórios Digitais Voltados ao Ensino. **Revista Brasileira de Informática na Educação**, v. 23, n. 02, p. 70, 2015.
- CAZELLA, S. C. et al. Recomendação de Objetos de Aprendizagem Empregando Filtragem Colaborativa e Competências. **Anais do Simpósio Brasileiro de Informática na Educação**, v. 1, n. 1, 2009.
- CHEN, H. **Personalized Learning Resources Recommendation Model Based on Transfer Learning**. Computer Science and Electronics Engineering. **Anais...2012**
- DRESCH, A.; LACERDA, D. P.; ANTUNES, J. A. V. Design Science Research. In: **Design Science Research: A Method for Science and Technology Advancement**. Cham: Springer International Publishing, 2015. p. 67–102.
- FLEISHMANN, A.; BASTOS, B. R.; PERNAS, H. **Sensibilidade à Situação em Sistemas Educacionais na Web**. [s.l.] Universidade Federal do Rio Grande do Sul, 2012.
- FREITAS, V. et al. Uma Arquitetura para Ecosystema de Software Científico. **Workshop em Desenvolvimento Distribuído de Software, Ecosystemas de Software e Sistemas-de-Sistemas**, p. 41–48, 2015.
- GAO, K.; ZHANG, B. **Modelling on clustering algorithm based on iteration feature**

- selection for micro-blog posts.** Modelling, Identification Control (ICMIC), 2014 Proceedings of the 6th International Conference on. **Anais...2014**
- JAWAHEER, G.; SZOMSZOR, M.; KOSTKOVA, P. Characterisation of Explicit Feedback in an Online Music Recommendation Service Categories and Subject Descriptors. **Analysis**, p. 317–320, 2010a.
- JAWAHEER, G.; SZOMSZOR, M.; KOSTKOVA, P. Comparison of implicit and explicit feedback from an online music recommendation service. **Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems - HetRec '10**, p. 47–51, 2010b.
- KITCHENHAM, B.; CHARTERS, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. **Engineering**, v. 2, p. 1051, 2007.
- LEI, K. et al. **A hybrid recommendation model based on the label propagation and VSM clustering.** Computer Science Education. **Anais...2012**
- LU, J. et al. Recommender system application developments: A survey. **Decision Support Systems**, v. 74, p. 12–32, 2015.
- LUCAS, J. P. et al. A hybrid recommendation approach for a tourism system. **Expert Systems with Applications**, v. 40, n. 9, p. 3532–3550, 2013.
- NERY, T. et al. BROAD Project: Semantic Search and Application of Learning Objects. **IEEE Technology and Engineering Education (ITEE)**, v. 7, n. 3, 2012.
- NÚÑEZ-VALDÉZ, E. R. et al. Implicit feedback techniques on recommender systems applied to electronic books. **Computers in Human Behavior**, v. 28, n. 4, p. 1186–1193, 2012.
- PEREIRA, C. K. et al. Extração de Características de Perfil e de Contexto em Redes Sociais para Recomendação de Recursos Educacionais. **Anais do Simpósio Brasileiro de Informática na Educação (SBIE)**, v. 25, n. Cbie, p. 506–515, 2014.
- PEREIRA, C. K. et al. Explorando Dados Ligados através de um Sistema de Recomendação Educacional. **Anais do Simpósio Brasileiro de Informática na Educação (SBIE)**, n. Sbie, p. No Prelo, 2015.
- QIAN, F. et al. Community-based user domain model collaborative recommendation algorithm. **Tsinghua Science and Technology**, v. 18, n. 4, 2013.
- REZENDE, P. A. A. et al. PERSONNA: proposta de ontologia de contexto e perfil de alunos para recomendação de objetos de aprendizagem. **Revista Brasileira de Informática na Educação**, v. 23, n. 01, p. 70, 2015.
- RIBEIRO, F. A. A.; FONSECA, L. C. C.; FREITAS, M. D. S. Recomendando Objetos de Aprendizagem a partir das hashtags postadas no Moodle. **XXIV Simpósio Brasileiro de Informática na Educação (SBIE 2013)**, v. 25, n. Cbie, p. 82–91, 2013.
- RUNESON, P. et al. **Case Study Research in Software Engineering. Guidelines and Example.** Wiley, Hob ed. [s.l: s.n.].
- SIMOES, L. et al. MMRRecommender: Metamodelo de Sistemas de Recomendação Aplicado a Grupos Educacionais. **Congresso Internacional de Informática Educativa (TISE)**, v. 12, n. November, p. 505, 2016.
- SIMOES, L. et al. Sistema de Recomendação de Serviços Baseado em uma Arquitetura Aberta para um Ecossistema de Software. **SBSI 2017 Proceedings of the annual conference**

on **Brazilian Symposium on Information Systems**, 2017.

TIAN, G. et al. Time-Aware Web Service Recommendations Using Implicit Feedback. **2014 IEEE International Conference on Web Services**, p. 273–280, 2014a.

TIAN, G. et al. Time-aware Web Service Recommendations Using Implicit Feedback. **EEE International Conference on Web Services Time-aware**, p. 273–280, 2014b.

ULLAH, F. et al. **Hybrid recommender system with temporal information**. Information Networking (ICOIN), 2012 International Conference on. **Anais...**2012

WEN, H. .; FANG, L. .; GUAN, L. . A hybrid approach for personalized recommendation of news on the Web. **Expert Systems with Applications**, v. 39, n. 5, p. 5806–5814, 2012.

WOHLIN, C. et al. **Experimentation in Software Engineering**. 1. ed. [s.l.] Springer, 2012a.

WOHLIN, C. et al. **Experimentation in Software Engineering Hardcover**. [s.l.] Springer Berlin Heidelberg, 2012b.

YANG, J.-M.; LI, K. F. **An Adaptive User-Genre-Item Model for Collaborative Filtering**. Communications, Computers and Signal Processing. **Anais...**2007

YIN, R. K. **Estudo De Caso - Planejamento E Metodos**. [s.l: s.n.].

ZHANG, Z. . B C; LIU, H. . C. Social recommendation model combining trust propagation and sequential behaviors. **Applied Intelligence**, v. 43, n. 3, p. 695–706, 2015.

APÊNDICE B

TRABALHOS PUBLICADOS E RELAÇÃO QUALIS

- ABDALLA, A. et al. R.ECOS - Educational Recommender Ecosystem. **Proceedings - 2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems, JSOS 2017**, n. May, p. 48–54, 2017. QUALIS B4
- SIMOES, L. et al. MMRecommender: Metamodelo de Sistemas de Recomendação Aplicado a Grupos Educacionais. **Congreso Internacional de Informática Educativa (TISE)**, v. 12, n. November, p. 505, 2016. QUALIS B5
- SIMOES, L. et al. Sistema de Recomendação de Serviços Baseado em uma Arquitetura Aberta para um Ecossistema de Software. **SBSI 2017 Proceedings of the annual conference on Brazilian Symposium on Information Systems**, 2017. QUALIS B2